Computer Aided Medical Procedures
Prof. Dr. Nassir Navab

Sebastian Walter Pölsterl

# Algorithms for Large-scale Learning from Heterogeneous Survival Data

Technische Universität München
Fakultät für Informatik

Technische Universität München
Fakultät für Informatik
Lehrstuhl für Informatikanwendungen in der Medizin

# Algorithms for Large-scale Learning from Heterogeneous Survival Data

*Sebastian Walter Pölsterl*

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des Akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

| | | |
|---|---|---|
| Vorsitzender: | | Univ.-Prof. Dr. Burkhard Rost |
| Prüfer der Dissertation: | 1. | Univ.-Prof. Dr. Nassir Navab |
| | 2. | Univ.-Prof. Dr. Torsten Hothorn<br>Universität Zürich, Schweiz |

Die Dissertation wurde am 19.01.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 18.04.2016 angenommen.

# Abstract

Many countries are nowadays challenged by ever-growing government expenditures for health care, which many seek to lower by employing electronic health records. Electronic health records systematically collect patients' past and current treatments with the aim at lowering administrative overhead and identifying inadequate treatments. Moreover, having access to large collections of clinical data creates an opportunity for clinical research. However, analyzing health records is often very challenging: first, they comprise a large and heterogeneous set of patient data, and second, they consist of variables collected from a wide range of sources, such as medications, allergies, biomarkers, medical images, and genetic markers – each of which offers a different partial view on a patient's state. Systematic analysis of such data is far beyond human capabilities and calls for machine learning techniques.

This thesis develops machine learning methods for predicting the time to an adverse event based on heterogeneous and high-dimensional health records. I introduce an improved training algorithm for the survival support vector machine that builds upon state-of-the-art methods in convex optimization to avoid the high time and space complexity of previous training algorithms. Experimental results on synthetic and real-world data demonstrate that my proposed optimization scheme allows analyzing datasets at least an order of magnitude larger than what would have been feasible with previous techniques. Second, I study dimensionality reduction methods in a comparative analysis of 19 feature extraction and feature selection methods. Whereas feature selection methods for learning from heterogeneous, high-dimensional feature vectors are well investigated, little work focused on feature extraction methods for survival analysis. I propose utilizing random survival forests to address two of the main problems encountered with feature extraction methods based on spectral embedding: 1) neighborhood graph construction and 2) out-of-sample extension. Experiments revealed that the proposed solution can represent similarities between patients better than the standard Euclidean distance and that feature extraction methods are a valuable alternative to feature selection methods, except if the number of available samples is low ($< 500$). Finally, I describe heterogeneous survival ensembles, which aggregate a wide range of survival models to leverage the diversity in available models. The success of such a model is evident by the fact that it was among the winning methods of the Prostate Cancer DREAM challenge.

**Keywords:** survival analysis, support vector machine, convex optimization, dimensionality reduction

# Zusammenfassung

Aktuell sind viele Länder mit stetig wachsenden Ausgaben für das Gesundheitssystem konfrontiert, was viele durch die flächendeckende Einführung von elektronischen Gesundheitsakten zu reduzieren versuchen. Die elektronische Gesundheitsakte eines Patienten vereint Information über bereits abgeschlossene und laufende Behandlungen mit dem Ziel administrative Mehrkosten zu reduzieren und unangemessene Behandlungen zu erkennen. Darüber hinaus fördert der Zugang zu vielen klinischen Daten auch die klinische Forschung. Allerdings ist die Analyse von Gesundheitsakten häufig sehr schwierig: erstens beschreiben sie viele unterschiedliche Patienten, und zweitens beinhalten sie eine hohe Anzahl an Indikatoren aus den unterschiedlichsten Bereichen, wie zum Beispiel Medikamente, Allergien, Blutwerte, radiologische und genetische Befunde, und so weiter. Eine systematische Analyse von derart komplexen Daten übersteigt die Fähigkeiten einer einzelnen Person und ist nur durch den Einsatz von Methoden aus dem Bereich des maschinellen Lernens zu bewältigen.

In dieser Arbeit werden Methoden des maschinellen Lernens vorgestellt, die es ermöglichen heterogene und hoch-dimensionale elektronische Gesundheitsakten zur Vorhersage der Überlebenszeit zu nutzen. Ich beschreibe einen verbesserten Algorithmus zum Trainieren einer *Survival Support Vector Machine* und greife dabei auf modernste Methoden zur Lösung von konvexen Problemen zurück; womit der hohe Zeit- und Speicherbedarf von existierenden Algorithmen vermieden wird. Experimente auf synthetischen und echten Daten zeigen dass dadurch erheblich größere Datensätze als mit vorangegangen Algorithmen analysiert werden können. Außerdem habe ich einen Vergleich von 19 Methoden zur Dimensionsreduktion durchgeführt. Wohingegen *Feature Selection* Methoden zum Lernen von heterogenen und hoch-dimensionalen Trainingsbeispielen gut untersucht sind, beschäftigten sich bisher wenige Arbeiten mit *Feature Extraction* Methoden zur Analyse von Überlebenszeiten. Ich nutze *Random Survival Forests* um zwei der Hauptprobleme von Feature Extraction Methoden basierend auf einer Eigenwertzerlegung zu umgehen: 1) die Erstellung des Nachbarschaftsgraphs, und 2) die Erweiterung zu bisher unbekannten Vektoren. Die Experimente haben gezeigt, dass das vorgeschlagene Verfahren Ähnlichkeiten zwischen Patienten besser darstellen kann als die gewöhnliche Euklidische Distanz und dass Feature Extraction Methoden nur dann eine wertvolle Alternative zu Feature Selection Methoden darstellen, wenn die Anzahl an Trainingsbeispielen ausreichend groß ist ($> 500$). Schließlich beschreibe ich *heterogeneous survival ensembles*, die die Vielzahl an vorhandenen Modellen zur Analyse von Überlebenszeiten ausnutzen, indem sie die Vorhersagen von mehreren Modellen zusammenführen. Der Vorteil dieses Modells ist anhand des siegreichen Beitrags zur *Prostate Cancer DREAM Challenge* erkennbar.

**Stichwörter:** Analyse von Überlebenszeiten, Support Vector Machine, Konvexe Optimierung, Dimensionsreduktion

# Acknowledgements

First of all, I would like to thank Prof. Nassir Navab for supporting me throughout my time as a PhD student. I am grateful that I always could rely on his unconditional support. I experienced the ups and downs of a PhD student, from “yes we can” to “we are all going to die.” It was in the latter times when Prof. Navab encouraged me to keep my head up, which enabled me to gain new strength and to focus again. Furthermore, I very much appreciate that he gave me the freedom to investigate topics I am personally interested in and which are not the primary focus of the chair.

I would like to thank Dr. Amin Katouzian for his enduring support in the scientific and in particular the non-scientific problems associated with research. His support was instrumental for completing my thesis.

During my four years at CAMP, I was lucky to work with the best colleagues one could wish for, which made work that more enjoyable: Martina Hilla, José Gardiazabal, Richard Brosig, Christian Schulte zu Berge, Loïc Peter, Christian Rupprecht, Benjamín Gutiérrez Becker, Ralf Stauder, Maximilian Baust, Tobias Lasser, Vasileios Belagiannis, Lichao Wang and Fausto Milletari. We shared countless interesting discussions, cakes, sweets, laughs and rounds of Schafkopf and Kicker. Christoph Graumann and Felix Bork deserve a special thank you for hosting me in Baltimore and showing me around such that I could leave the town alive.

In addition, I would like to thank Felix Buggenthin for the regular “therapy” sessions, which made life as a PhD student bearable, because I knew that others are suffering too.

Finally, the biggest thank you goes to my family; without them, nothing of what you are going to enjoy below would have been possible – their support is admirable. The gave me the confidence and freedom to always pursue what interested me the most, which I enjoy to this date. I tremendously enjoy our regular get-togethers, which offer a break from daily routine.

# Contents

Notations ix

1 Introduction 1
- 1.1 Historical Overview . . . 1
- 1.2 About this Thesis . . . 6

2 Survival Analysis 11
- 2.1 Censoring . . . 11
- 2.2 Truncation . . . 13
- 2.3 Functions of Survival Time . . . 14
- 2.4 Non-parametric Estimators . . . 20
- 2.5 Hypothesis Testing . . . 25

3 Predictive Models for Survival Analysis 31
- 3.1 Accelerated Failure Time Model . . . 31
- 3.2 Cox's Proportional Hazards Model . . . 35
- 3.3 Support Vector Machines for Survival Analysis . . . 46
- 3.4 Boosting . . . 60
- 3.5 Tree-based Methods for Survival Analysis . . . 71
- 3.6 Random Survival Forest . . . 79
- 3.7 Evaluation Measures . . . 83

4 Missing Data 91
- 4.1 Missing Data Generating Processes . . . 92
- 4.2 Multiple Imputation . . . 93

5 Fast Training of Survival Support Vector Machine 99
- 5.1 Analysis of Existing Approaches . . . 99
- 5.2 Survival Support Vector Machine with Squared Hinge Loss . . . 101
- 5.3 Truncated Newton Optimization . . . 103
- 5.4 Survival Analysis as Regression Problem . . . 111
- 5.5 Hybrid Model . . . 112
- 5.6 Non-linear Extension . . . 113
- 5.7 Comparison of Survival Support Vector Machines . . . 117

**6 Survival Analysis For High-Dimensional, Heterogeneous Medical Data 127**
6.1 Dimensionality Reduction Methods . . . . . 127
6.2 Multiview Spectral Embedding For Survival Analysis . . . . . 132
6.3 Evaluation of Feature Extraction and Feature Selection Method . . . . 139

**7 Predicting Survival of Prostate Cancer Patients 155**
7.1 Prostate Cancer DREAM Challenge . . . . . 155
7.2 Extracting Features Describing Patient Records . . . . . 158
7.3 Considerations Regarding Prostate Cancer DREAM Challenge . . . . . 164
7.4 Heterogeneous Survival Ensembles . . . . . 164
7.5 Cross-Validation Results . . . . . 169
7.6 Results of Prostate Cancer DREAM Challenge . . . . . 174

**8 Conclusion 187**

**A Experiments Regarding Dimensionality Reduction Methods 191**
A.1 Hyper-Parameter Configurations . . . . . 191
A.2 Description of Datasets . . . . . 192
A.3 View-specific Coefficients for Framingham Offspring Dataset . . . . . 196

**B Prostate Cancer DREAM Challenge Data 199**
B.1 Features . . . . . 199
B.2 Hyper-Parameter Configurations . . . . . 209

**Bibliography 211**

**Author's Publications 235**

# Notations

Throughout this thesis I will use the following notations. Scalars, vectors, matrices and sets are denoted by lower case letters, bold face lower case letters, bold face capital letters and calligraphic capital letters, respectively. The $i$-th row of a matrix $\boldsymbol{M}$ is denoted by $\boldsymbol{M}_i$ and the value in the $i$-th row and $j$-th column as $\boldsymbol{M}_{i,j}$ or $(\boldsymbol{M})_{i,j}$. Similar for a vector $\boldsymbol{v}$, where $v_i$ and $(\boldsymbol{v})_i$ denote the $i$-th element.

$T$ and $C$ denote non-negative random variables representing the survival time and time of censoring, respectively. Concrete values for the survival time and censoring time are denoted by $t > 0$ and $c > 0$, respectively, and may have a subscript to indicate patient-specific survival and censoring times.

For a survival model, the training set $\mathcal{D}$ consists of $n$ triplets $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^n$, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^\top \in \mathbb{R}^p$ is a $p$-dimensional feature vector, $y_i = \min(t_i, c_i)$ is the observed time, and $\delta_i = I(t_i \leq c_i)$ an indicator whether $y_i$ corresponds to a survival time or time of censoring. Samples in $\mathcal{D}$ can also be described as the matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$, and the vectors $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ and $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)^\top$. Estimates of quantities or functions are indicated by a *hat*, in particular, $\hat{f}(\boldsymbol{x})$ is the prediction of feature vector $\boldsymbol{x}$ based on model $\hat{f}$.

An overview of notations is available from the table below.

| Notation | Description |
|---|---|
| $I(\cdot)$ | Indicator function. |
| $E(\cdot)$ | Expected value of a random variable. |
| $\mathrm{Var}(\cdot)$ | Variance of a random variable. |
| $P(\cdot)$ | Probability. |
| $O(\cdot)$ | Asymptotical upper bound on an algorithm's time/space requirements. |
| $C$ | Non-negative random variable denoting the time of censoring. |
| $T$ | Non-negative random variable denoting the survival time. |
| $Y$ | Random variable denoting the observed time: $Y = \min(T, C)$. |
| $c_i$ | Time of censoring of $i$-th individual. |
| $t_i$ | Survival time of $i$-th individual. |

Continued on next page

| Notation | Description |
|---|---|
| $y_i$ | Observed time of $i$-th individual: $y_i = \min(t_i, c_i)$. |
| $\delta_i$ | Event indicator of $i$-th individual: $\delta_i = I(t_i \leq c_i)$. |
| $\omega_i$ | Inverse probability of censoring weight for $i$-th subject. |
| $\mathcal{D}$ | Survival data from $n$ patients: $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^{n}$. |
| $\mathcal{R}_i$ | Risk set at time point $t_i$: $\mathcal{R}_i = \{j \mid y_j \geq t_i\}$. |
| $h(t)$ | Hazard function. |
| $H(t)$ | Cumulative hazard function. |
| $F(t)$ | Cumulative distribution function. |
| $S(t)$ | Survival function. |
| $\mathbb{N}$ | Set of all natural numbers. |
| $\mathbb{R}$ | Set of all real numbers. |
| $\mathbb{Z}$ | Set of all integer numbers. |
| $\varnothing$ | Empty set. |
| $\mathbb{1}_m$ | Vector of all ones of size $m$. |
| $\mathbf{0}_m$ | Vector of all zeros of size $m$. |
| $\boldsymbol{I}_m$ | Identity matrix of size $m \times m$. |
| $\boldsymbol{v}^\top$; $\boldsymbol{M}^\top$ | Transpose of vector $\boldsymbol{v}$ or matrix $\boldsymbol{M}$. |
| $v_i$; $(\boldsymbol{v})_i$ | Value of $i$-th element of vector $\boldsymbol{v}$. |
| $\boldsymbol{M}_i$ | Vector corresponding to $i$-th row of matrix $\boldsymbol{M}$. |
| $\boldsymbol{M}_{i,j}$; $(\boldsymbol{M})_{i,j}$ | Value in the $i$-th row and $j$-th column of matrix $\boldsymbol{M}$. |
| $\mathrm{diag}(\boldsymbol{v})$ | Diagonal matrix with diagonal entries $v_1, \ldots, v_k$. |
| $\mathrm{tr}(\boldsymbol{M})$ | Trace of matrix $\boldsymbol{M}$. |
| $\langle \cdot, \cdot \rangle$ | Inner product in Euclidean space. |
| $\lVert \cdot \rVert_p$ | $\ell_p$ norm of a vector: $\lVert \boldsymbol{x} \rVert_p = (\sum_{j=1} \lvert \boldsymbol{x}_j \rvert^p)^{1/p}$. |
| $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ | Inner product in Hilbert space $\mathcal{H}$. |
| $\lVert \cdot \rVert_{\mathcal{H}}$ | Norm in Hilbert space $\mathcal{H}$. |
| $k(\boldsymbol{x}, \boldsymbol{z})$ | Kernel function. |
| $\boldsymbol{K}$ | Kernel matrix with elements $\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. |
| $[a; b]$ | Closed interval from $a$ (included) to $b$ (included). |
| $[a; b[$ | Half-open interval from $a$ (included) to $b$ (excluded). |
| $]a; b]$ | Half-open interval from $a$ (excluded) to $b$ (included). |

# 1 Introduction

## 1.1 Historical Overview

Before 1538 no information about a city's or country's population were systematically recorded. In that year, Thomas Cromwell (1485-1540) ordered that every parson maintains a parish register that records every baptism, marriage and burial within a priest's parish [256]. Based on recordings of parishes in and around the City of London, James I. (1566-1625) mandated at the beginning of the 17th century that the Company of Perish Clerks publishes weekly statistics about the number of deaths and their causes [215]. The main purpose of these so called *bills of mortality* was to monitor the outbreak of the plague in London. In 1662, John Graunt (1620-1674) analyzed data from bills of mortality from the years 1629-1636 and 1647-1658 to produce the first ever life table [16, 118, 139]. Graunt's life table allowed inferring crude estimates of child mortality, population size and life expectancy, which makes Graunt arguably the founder of demography [139]. However, Graunt's estimates were flawed in several ways: 1) he had no information about the age at death, because bills of mortality only recorded the cause of death, 2) he was only provided with the number of burials rather than the actual number of deaths (similarly for baptized children as proxy for newborns), and 3) he did not provide a detailed description of the methods he used to arrive at his life table [16]. Graunt was mostly aware of these shortcomings, in fact, he referred to his own work as "hav[ing] reduced several great confused volumes into a few perspicuous tables, and abridged such observations as naturally flowed from them, into a few succinct paragraphs, without any long series of multiloquious [*sic*] deductions" [24, p. 4].

Many of the flaws in Graunt's data were absent in data from the city of Breslau, analyzed by Edmond Halley[1] (1656–1742) [123]. The data contained records of births and deaths for the years 1687–1691. In contrast to London's population, which was under constant changes due to migration, Breslau's population at that time remained approximately constant and the data contained information about a person's sex and age at death [16, 58]. Halley [123] published his conclusions in 1693 (reprinted in [140]), where he presented, among others, a more accurate life table. Based on his life table, he argued that the government should adjust the price of life annuities by relating their

[1] Halley's main profession was astronomy, Halley's comet was named in his honor.

price to a person's age[2] [16, 58] – a finding that is still reflected in today's insurance policies.

In the first half of the 19th century, physicians commonly believed that diseases were caused by non-specific inflammations of organs and most diseases could be treated by applying leeches to the area of the corresponding organ [216]. Pierre Charles Alexandre Louis (1787-1872) opposed this interpretation and called for a new school of thought that derived conclusions by systematically recording information obtained from observing patients and analyzing specimens from autopsies. Subsequently, his "numerical method" would be used to analyze the data and provide evidence about the effectiveness of treatment. It consisted of dividing subjects into groups, calculating a statistic for each group (usually the mean of a measurement) and comparing its value among the groups [13]. Louis [200, p. 59] described his reasoning as follows:

> "In any epidemic, for instance, let us suppose five hundred of the sick, taken indiscriminately, to be subjected to one kind of treatment, and five hundred others, taken in the same manner, to be treated in a different mode; if the mortality is greater among the first, than among the second, must we not conclude that the treatment was less appropriate, or less efficacious in the first class, than in the second?"

An often cited application of his "numerical method" is his study on the effects of blood-letting [199, 200]. It was based on 78 patients treated for pneumonia, of which 28 had died [200, p. 2]. He created a table that compared duration of disease to time of first bleeding and total number of bleedings. He formed two roughly equal sized groups: the first group comprising patients who have been treated during the first four days after onset, and a second group comprising patients who have been treated five to nine days after onset. Louis's results revealed that 18 of 41 patients (44%) died in the first group and that 9 of 36 patients (25%) died in the second group, whereas the mean duration of pneumonia was 17.8 and 20.8 days among the survivors, respectively [200, pp. 9 & 5]. He concluded that blood-letting in the early stages of pneumonia had little effect on duration, and that blood-letting should be limited to severe cases, where the risk/benefit ratio is more favorable [200, pp. 48 & 49]. The work by Louis is considered as the birth of evidence-based medicine [216].

Nineteenth-century England was smitten with several epidemics such as smallpox and cholera, which many physicians believed to be transmitted by "bad air" (miasma theory). Against popular belief, John Snow (1813-1858) was convinced that germs and unhygienic conditions were the main causes [184, 273]. In particular, Snow [272] showed that the cholera outbreak in Soho, London, in 1854 could be traced back to a single water pump supplied with polluted water [184]. Snow summarized the

[2] At that time, a life annuity provided a yearly payment to its buyer, independently of age. People who acquired an annuity with young age would receive more money overall, which increased the financial risk for its seller.

location of deaths around several water bumps in London in a map to reject the null hypothesis that death is equally likely in each area – which is now considered the first epidemiological study [184].

At the same time, William Farr (1807-1883), who is another key figure in epidemiology and public health, opposed Snow's germs theory and tried to proof that cholera is transmitted by air. He conceived a mathematical model that inversely relates a district's elevation above the high water mark of the Thames to the mortality rate, based on data from 53,293 cholera deaths in 1849 [92]. He justified his model by arguing that organic materials accumulate in the water, degrade and release an "aqueous vapour [*sic*]" [92, p. 163] that is most concentrated in the coastal districts and evaporates before it reaches higher districts. In Farr's report on the 1866 cholera outbreak, he reversed his position on the transmission of cholera and aligned himself with Snow's reasoning [93]. Moreover, Farr went beyond analyzing data retrospectively by noting that epidemics follow a "principle of periodicity" [94, p. 316], which he leveraged by fitting a third degree polynomial to a dataset consisting of the number of deaths at ten time points [184]. He used his knowledge about the "[l]aws of [e]pidemics" [94, p. 317], as he called them, to predict the decline of the 1865 cattle plague [184] and the surge of cholera in 1854 [94, p. 359].

Although seminal works of Louis, Snow, Farr and others were met with skepticism and denial first, by the beginning of the 20$^{\text{th}}$ century, the belief that medical research ought to be based on systematic collection of data and statistical analyses prevailed. However, an important aspect of today's clinical trials was still absent: randomization. The concept of randomization in clinical research was popularized by Austin Bradford Hill (1897-1991) in a series of papers in the *Lancet* [142]. Eventually results of the first randomized clinical trial on streptomycin in pulmonary tuberculosis were published in 1948 [1]. Ronald Aylmer Fisher (1890-1962) introduced several key concepts used by Hill in his book "The Design of Experiments" [100]. Fisher mostly studied experiments in agriculture and made numerous contributions to statistics: maximum likelihood estimation [97], analysis of variance [98], and Fisher's exact test [96, 98], just to name a few.

In this thesis, the focus is on learning predictive models from right censored samples. If a sample is censored, it is only known that an event occurred in a particular (possibly infinite) time interval, but not its exact time (see section 2.1). If areas of the sampling distribution cannot be observed at all, data are truncated. The problem of truncated samples was first recognized by Francis Galton (1822-1911) in his 1897 study on the speed of American trotting horses [107]. Galton analyzed lap times of horses around a one mile course, which were only recorded for horses requiring less than 150 seconds. He estimated the mean by assuming a right truncated normal distribution, because he was only provided with samples below the truncation point of 150 seconds [59, p. 2]. Galton's estimator was rather ad hoc; better techniques for estimating the parameters of a normal distribution from truncated samples were later proposed by Fisher [99], Pearson [229], and Pearson and Lee [230]. Stevens [276] (in an appendix to [30]) first considered estimating parameters of a normal distribution from truncated samples with known truncation time and number of unmeasured observations, which today is better known as *censored* observations – the term “censoring” was first used by Hald [122].

Up to the mid-20th century, most statistical methods in clinical research treated a disease and its associated outcome as a binary event (e.g. survival to 12 months from entry into the trial). Therefore, proving the efficacy of a treatment usually consisted of performing Fisher’s exact test or a $\chi^2$ test. More information about a disease can be inferred when directly studying the time until an event of interest occurs, which is the objective in *survival analysis*. Examples from the medical domain are the time until death, until onset of a disease, or until pregnancy. For instance, by assuming a normal distribution for the time until an event, its parameters could be estimated from censored observations by employing techniques of Stevens [276] mentioned above. However, the normal distribution is often unsuitable for biomedical data, because it does not describe the distribution of event times well. Alternative distributions were extensively studied in the 1930s to describe the life cycle of materials, mechanical and electronic systems for reliability analysis. The exponential distribution and its generalization the Weibull distribution originated from research in this area [312, 313]. Epstein and Sobel [88] described a technique to estimate the parameter of an exponential distribution from censored observations. The first non-parametric estimator for survival analysis was proposed by Kaplan and Meier [167]: the well-known estimator of the survival function (see section 2.4).

Estimators mentioned in the previous paragraph only work reliably if the patient population is homogeneous and the survival distribution is identical for all patients in a study. For instance, the assumption would be violated if survival of overweight patients is shorter than for patients with normal weight. Regression models address this problem by incorporating one or more patient characteristics in the estimation of the survival distribution. The development of such models unfolded similar to the estimators of the survival function in the previous section. Early regression models required to explicitly specify the form of the survival distribution (e.g. exponential distribution), which allowed employing maximum likelihood estimation [110, 331]. However, the true underlying distribution function is usually unknown and choosing a suitable parametric form is considered an art. Eventually, Cox [67] proposed the first semiparametric regression model that could be applied to censored survival data without explicitly specifying the survival distribution (see section 3.2).

Advancements in survival analysis in the 1960s and 1970s, in particular with respect to regression models, were fostered by spreading of computer technology, which allowed analyzing more complex data without the burden of manual calculus. Moreover, the advent of computers attended the birth of machine learning and artificial intelligence. The first machine learning program was completed by Arthur Lee Samuel in 1955 [314], but was not published until four years later [249]. Samuel developed a program to play checkers that was able to learn from playing against other instances of the program or against human players. Around the same time, Frank Rosenblatt proposed the perceptron, which is a simple linear binary classifier [242]. By organizing multiple perceptrons into a network, Rosenblatt [242] showed that multilayer perceptrons can solve complex problems. Eventually, machine learning became an entity on its own, distinct from artificial intelligence. The fields started to drift apart in the 1980s, when research in artificial intelligence was mainly focusing on emulating humans' way of thinking (cognitive simulation), rather than assisting them in learning from examples (inductive learning) [186, 266]. Langley [186, p. 277] defined machine learning as "the study of any methods that improved performance with experience." Therefore, pattern recognition and data mining are closely related fields.

In the late 1990s and early 2000s advancements in DNA sequencing [155] resulted in a flood of new data for which existing survival models, such as Cox's proportional hazards model, were inadequate. Rosenwald *et al.* [243] studied the survival of 240 patients after chemotherapy for diffuse large-B-cell lymphoma based on 7,399 expression levels representing approximately 4,128 genes. The example illustrates that the number of features greatly exceeds the number of samples, in which case traditional survival models cannot be applied. Around the same time research in machine learning expanded into the field of statistics [186], and vice versa, which set the stage for machine learning techniques to tackle these problems.

Today, many successful ideas from machine learning have been adapted for survival analysis; extensions of boosting, random forest, and support vector machine are discussed in chapter 3. Nevertheless, early work in survival analysis remains relevant even today. According to a 2014 report in *Nature* [303] the impact of the work of Kaplan and Meier [167] and Cox [67] continuous to be massive: they are considered the two most cited papers in the field of statistics with more than 38,000 and 28,000 citations, respectively. This not only shows the impact computers had – and still have – on the conversion of theoretical results to real-world applications, but also the significance of survival analysis as a whole today.

## 1.2 About this Thesis

### 1.2.1 Motivation

More and more national governments are committed to integrating the use of electronic health records (EHR) to improve quality of care and reduce health care costs [3]. Hence, massive amounts of medical data are collected every day, which are far beyond what a human could analyze. However, in order to improve patients' outcomes and lower costs, it is necessary to identify existing problems in patient care and to resolve them, which is only reasonable through machine learning techniques (see e.g. [161]). In the United States of America, the Medicare and Medicaid EHR Incentive Program requires physicians to make "meaningful use" of electronic health records, which, among others, includes consolidating a clinical decision support system [52]. In 2013, an estimated 69% of office-based physicians in the USA participated in the "meaningful use" program [151], which exemplifies the demand for such systems.

The primary interest in the medical domain is the analysis of time until an adverse event occurs (*survival analysis*). Thus, the objective of survival analysis is to examine how a particular set of covariates affects the time until a patient is going to experience a particular event, such as death or reaching a specific state of disease progression. In order to leverage electronic health records for survival analysis, a survival model must be apt to

1. a very large sample size,
2. a large set of features, of which only an unknown subset actually affects the time of an event,
3. feature vectors that are a mix of continuous, categorical and ordinal attributes,
4. incomplete feature vectors (missing values), and
5. censored event times.

### 1.2.2 Aims

The aim of this dissertation is to propose novel algorithms for learning from large datasets of right censored observations with high-dimensional and heterogeneous feature vectors. Consequently, the primary focus is on the first three requirements stated above.

*Note.* The focus of this dissertation is solely on data acquired from the medical domain. Nevertheless, as censoring is often encountered in reliability analysis [10] and economics [213], my contributions are immediately applicable to a wide range of applications outside of medicine.

### 1.2.3 Contributions

This dissertation makes several important contributions to learning from survival data:

1. I propose a novel training algorithm of linear survival support vector machines (SVMs) with ranking constraints that leverages truncated Newton optimization of the primal objective function and ordered statistics tree to lower the complexity of training from $O(pq_e^2 n^4)$ to

   $$[O(n \log n) + O(np + p + n \log n)] \cdot \bar{N}_{\text{CG}} \cdot N_{\text{Newton}},$$

   where $q_e$ is the percentage of uncensored records, $n$ the number of samples, $p$ the number of features, $\bar{N}_{\text{CG}}$ the average number of conjugate gradient iterations, and $N_{\text{Newton}}$ the total number of Newton updates (see chapter 5).

2. I extend the training algorithm for linear survival SVMs with ranking constraints to non-linear decision functions in chapter 5. I show that it is possible to leverage techniques used in learning a linear model by directly applying the representer theorem to the primal objective function of a non-linear model.

3. In chapter 5, I describe a hybrid survival SVM that combines two loss functions: 1) the squared hinge loss used in survival SVMs with ranking constraints, and 2) the squared loss used in regression.

4. A novel algorithm to construct neighborhood graphs from heterogeneous survival data is presented in chapter 6. It leverages random survival forests to account for censoring and high-dimensional, heterogeneous feature vectors. Experimental results show that the proposed approach preserves local neighborhoods considerably better than using the common Euclidean distance, which is unsuitable if feature vectors are a mix of continuous and categorical attributes.

5. In chapter 6, I describe the results of a large empirical study on evaluating the performance of 10 combinations of feature extraction methods and 8 survival models with and without embedded feature selection on three clinical datasets. To the best of my knowledge, this is the first study with focus on survival analysis that thoroughly investigates in which situations feature selection and feature extraction methods excel.

6. Heterogeneous survival ensembles comprise base learners that differ in the objective function that is optimized, and hence offer a greater diversity. They are proposed in chapter 7 in the context of my winning solution to the Prostate Cancer DREAM Challenge.

The majority of ideas presented in chapter 5 can be found in

> S. Pölsterl *et al.*, "Fast training of support vector machines for survival analysis," in *Machine Learning and Knowledge Discovery in Databases*, A. Appice *et al.*, Eds., ser. Lecture Notes in Computer Science, 2015, pp. 243–259. DOI: 10.1007/978-3-319-23525-7_15.

Methods and results presented in chapter 6 have been submitted and were still under review at the time of submission of this thesis:

> S. Pölsterl *et al.*, "Survival analysis for high-dimensional, heterogeneous medical data: Exploring feature extraction as an alternative to feature selection," *Artificial Intelligence in Medicine*, 2016, submitted.

At the time of submission of my dissertation, the publication on heterogeneous survival ensembles proposed in chapter 7 was under preparation and is going to appear in the DREAM Challenges channel at *F1000Research*[3] once the overview paper of the Prostate Cancer Challenge has been submitted to *Nature Biotechnology*.

### 1.2.4 Outline

The contents of this thesis are mostly self-contained, therefore, I will start defining basic concepts and quantities used in survival analysis in chapter 2. In sections 2.1 and 2.2 (pages 11 and 13), I will define different types of censoring and truncation and illustrate these concepts using examples. A statistical framework that is commonly used to describe survival data in the form of survival function, hazard function, and cumulative hazard function is given in section 2.3.

Next, I will define well-known non-parametric estimators of survival function and cumulative hazard function for right censored as well as truncated event times in section 2.4 (page 20). I will conclude chapter 2 by presenting a class of statistical tests to assess whether the distribution of survival times between two or more groups of patients differ significantly (section 2.5 on page 25).

Based on basic concepts and methods specified in chapter 2, I will describe the most important models for analyzing survival data with right censored event times in chapter 3 (page 31). The accelerated failure time model and Cox's proportional hazards model are traditional statistical models and are explained in sections 3.1 and 3.2, respectively (pages 31 and 35).

Section 3.3 illustrates different approaches to adapting survival support vector machines for survival analysis by casting it as learning-to-rank problem (section 3.3.1 on page 46), a regression problem (section 3.3.2 on page 48), or a quantile regression problem (section 3.3.4 on page 51). Initially, I will define the models in form of a linear decision

[3] http://f1000research.com/channels/DREAMChallenges

function, but the extension to non-linear decision functions can be traced back to ideas described in section 3.3.5 (page 53).

A formal definition of gradient boosting and several adaptations to survival analysis are discussed in section 3.4 (page 60). Gradient boosting methods are ensemble methods that are characterized by the choice of base learner (section 3.4.3 on page 63) and loss function they optimize (section 3.4.4 on page 68).

Section 3.5 (page 71) covers survival trees and reviews split criteria that have been proposed in the past. This forms the basis for section 3.6 (page 79), where ensembles of survival trees, namely random survival forests are defined.

Methods to estimate the performance of survival models on right censored test data are discussed in section 3.7.

For the sake of completeness of this thesis, chapter 4 briefly covers general methods to address the missing value problem, in particular by Multivariate Imputation using Chained Equations (MICE). After reviewing the basics of survival analysis and state-of-the-art survival models in chapters 2 to 4, I will present my main contributions in chapters 5 to 7.

I will present an improved optimization algorithm for three types of survival support vector machines in chapter 5: 1) ranking-based, 2) regression-based, and 3) a combined ranking and regression approach. I will demonstrate that the proposed optimization scheme of linear ranking-based survival support vector machine lowers computational costs of training by minimizing the primal objective function (section 5.2 on page 101) and combining truncated Newton optimization with order statistic trees (section 5.3 on page 103). An extension to non-linear decision functions that utilizes the same optimization scheme is proposed in section 5.6 (page 113). In section 5.7 (page 117), results on synthetic and real-world datasets will demonstrate the superiority of my proposed optimization scheme over existing training algorithms, which fail due to their inherently high time and space complexities when applied to large datasets.

Chapter 6 focuses on feature extraction algorithms for high-dimensional, heterogeneous medical data. First, I will briefly review related work in feature selection and feature extraction in section 6.1 (page 127). Next, I will focus on spectral embedding algorithms, in particular multiview spectral embedding, which considers that features have different statistical properties when determining a low-dimensional representation of the training data. In section 6.2 (page 132), I will propose using random survival forests to accurately determine local neighborhood relations from right censored survival data consisting of high-dimensional, heterogeneous feature vectors. I evaluated 10 combinations of feature extraction methods and 8 survival models with and without intrinsic feature selection in the context of survival analysis on three clinical datasets. Results in section 6.3 (page 139) demonstrate that survival models with embedded feature selection (random survival forest and gradient boosted models) outperform feature extraction methods, because they are unable to reliably identify the underlying manifold, which makes them of limited use in these situations. For large sample sizes, feature extraction methods perform as well as feature selection methods.

Chapter 7 will cover my contributions to the Prostate Cancer DREAM Challenge, where the objective was to predict survival of patients with metastatic, castrate-resistant prostate cancer from a patient's health record. After providing an overview of the challenge's tasks and data in section 7.1 (page 155), I will present my approach to extracting information from health records in section 7.2 (page 158). A novel ensemble technique to combine several survival models, each optimizing a different loss during training, is proposed in section 7.4 (page 164). Preliminary results and insights obtained before the final submission to the Prostate Cancer DREAM Challenge are illustrated in section 7.5 (page 169). The final results of the challenge in section 7.6 (page 174) demonstrate that using a heterogeneous ensemble of survival models outperformed competing methods.

Finally, I will close my dissertation with concluding remarks in chapter 8.

# 2 Survival Analysis

## 2.1 Censoring

Most clinical studies enroll patients during a fixed period of time and then follow these patients for a certain amount of time. During the study period, patients are asked to complete one or more follow-ups with the purpose to record how a patient's health changed over time. For instance, the Framingham Offspring study [166] started to enroll participants in 1971 with the purpose to identify common factors and characteristics that contribute to cardiovascular disease. Rather than studying the pathogenesis of the disease, the primary motivation for the Framingham Offspring study was to determine the difference between people who developed cardiovascular disease and those that remained disease-free. Therefore, participants were followed for several years and if a participant experienced a cardiovascular event, such as coronary heart disease, angina pectoris, or stroke, the exact time of the event was recorded. In the beginning, 5,124 patients enrolled in the study and, as of 2014, nine follow-up exams – approximately four years apart from each other – were conducted.

However, not all participants enrolled at the same date nor did they perform their follow-up exams simultaneously. In addition, some patients decided to leave the study or simply were unreachable, which means no information beyond the time of their last follow-up is available. A graphical representation of this situation is depicted in fig. 2.1: Patient A was lost to follow-up after three months with no recorded cardiovascular event, patient B experienced an event four and a half months after enrollment, patient D withdrew from the study two months after enrollment, and patient E did not experience any event before the study ended. Consequently, the exact time of a cardiovascular event could only be recorded for patients B and C; their records are *uncensored*. For the remaining patients it is unknown whether they did or did not experience an event after termination of the study. The only valid information that is available for patients A, D, and E is that they were event-free up to their last follow-up. Therefore, their records are *censored*.

Three distinct patterns of censoring exist: right censoring, left censoring, and interval censoring. Figure 2.1 illustrates *right censoring*, which has been explained above. A patient record is *left censored* if an event occurred prior to a specific time point $t$, but its exact time of occurrence is unknown. *Interval censoring* occurs when a patient experienced an event at an unspecified time point between times $t_1$ and $t_2$.

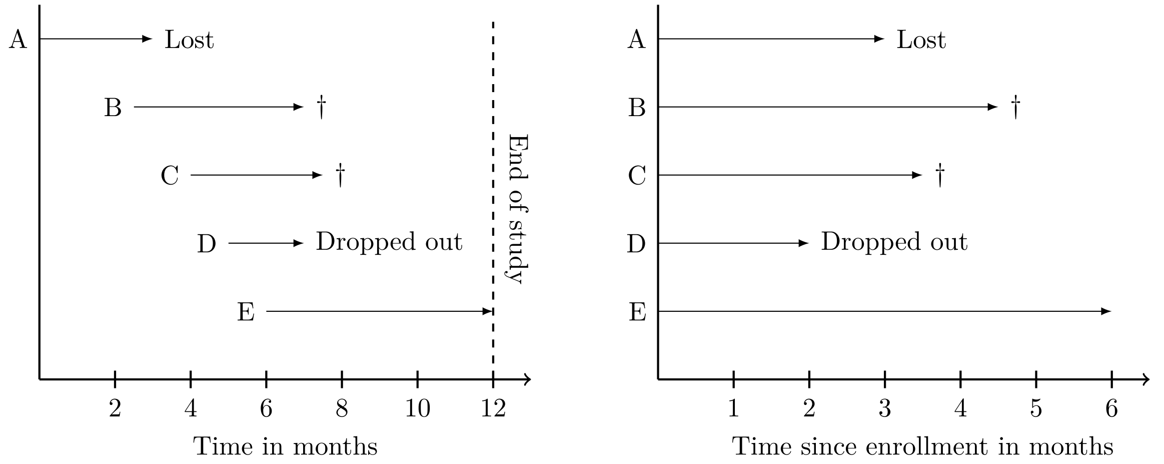


**Figure 2.1**: Example of right censoring in a clinical study. Patients B and C experienced an event during the study period and their records are *uncensored*. Records of patients A, D, and E are *right censored* because they did not experience an event until they left the study (A and D) or the study ended (E). A cross represents occurrence of an event.

Formally, each patient record consists of a set of covariates $\boldsymbol{x} \in \mathbb{R}^d$, and the time $t > 0$ when an event occurred or the time $c > 0$ of censoring. Since censoring and experiencing and event are mutually exclusive, it is common to define an event indicator $\delta \in \{0; 1\}$ and the observable survival time $y > 0$. The observable time $y$ of a right censored sample is defined as

$$y = \min(t, c) = \begin{cases} t & \text{if } \delta = 1, \\ c & \text{if } \delta = 0, \end{cases}$$

where $\delta = 1$ if a patient experienced and event and zero otherwise. For left censored observations, min is replaced by max: $y = \max(t, c)$. If a record is subject to left and right censoring at times $c_{\text{left}}$ and $c_{\text{right}}$, respectively, data can be represented in a similar way by extending the definition of $\delta$ to $\delta \in \{-1; 0; 1\}$, where $-1$ denotes a left censored time, 0 a right censored time, and 1 an event. The observable time $y$ is defined as $y = \max(\min(t, c_{\text{left}}), c_{\text{right}})$. Such a record is called *doubly censored*.

In clinical studies, the most common type of censoring is right censoring and thus I will focus on right censoring for the remainder of this thesis. Generally, there are three mechanisms that can lead to right censored records. *Type I censoring* refers to a study where all $n$ subjects are enrolled at a specific time point $t_{\text{start}}$ and are followed until time point $t_{\text{end}}$. Individuals that did not experience an event up to time $t_{\text{end}}$ are censored at that exact time point. Consequently, the censoring time is fixed for all participants: $c_i = t_{\text{end}}$ $(\forall i = 1, \ldots, n)$. The generalized form of type I censoring refers to the scenario where each individual has its own starting time, but the end of the study is predetermined and thus each participant has its own predetermined time of censoring.

The time of censoring in a study under *type II censoring* is not predetermined by the duration of the study but a fixed number of events: the study is terminated if $k$ of $n$ participants ($k < n$) experienced an event. Therefore, $c_i = t_k$ for all patients who remained event-free up to time $t_k$, which means that the time of censoring is random.

Finally, *competing risk censoring* arises if a patient is affected by a competing event that results in his or her removal from the study. Hence, it is impossible to observe the actual event of interest anymore. For instance, if the primary event of interest is death from heart disease, patients who died from cancer or from any other cause are right censored due to competing risks. To allow inference, one assumes that the event time and time of censoring are independent of each other, which is termed *non-informative censoring*. This assumption would be violated if most patients dropping out of a study would experience an event shortly after they left the study, which would lead to biased estimates of survival time, because individuals that can still be observed are no longer representative of the overall population. If the nature of competing risks is unknown, one usually assumes *random censoring*, which is a simplified version of competing risk censoring. Figure 2.1 shows an example where patients A and D randomly drop out of the study due to unknown reasons such as accidental death or moving to another area. Moreover, the figure indicates that most clinical studies are affected by both random censoring (patients A and D) as well as type I censoring (patient E) [176, p. 70].

Despite the differences between the three mechanisms described above, data collected during a study with $n$ participants can be summarized by the set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^n$ for all of them.

## 2.2 Truncation

Another property of survival data closely related to censoring, yet with distinct differences, is *truncation*. Truncated data arises when part of the overall population cannot be observed at all and thus no information about this part of the population is available. In contrast, censored records at least contain partial information that can be used to drive inference. Ignoring truncation during inference would lead to biased estimates, therefore special care has to be taken when analyzing data with truncated survival times. Similar to censoring, times can be left or right truncated. They are left truncated if a study excludes subjects experiencing an event prior to a particular time of truncation. Right truncation arises if observations, whose event times exceed a specified time of truncation, are excluded. Moreover, data truncation does not exclude censoring, in fact, it is common that left truncated data contains right censored observations [176, p. 71].

**Example 2.1.** A classical example for left truncation (also called *late* or *delayed entry*) is a study on the survival of members of the Channing House retirement community in Palo Alto, California [156]. Members of the community had access to a health care program that provided them with easy access to medical care without increasing their financial burden. Survival times in this data are left truncated, because people needed to reach a certain age before getting admitted. Consequently, individuals who died at an earlier age were systematically excluded from the study, resulting in overestimated survival probabilities if left truncation is ignored.

**Example 2.2.** As an example for right truncation, consider a study regarding incubation periods for acquired immune deficiency syndrome (AIDS) patients, who were infected with the human immunodeficiency virus (HIV) by contaminated blood transfusions [182]. The primary interest in the study was to investigate differences in incubation time, i.e., the time between infection and AIDS diagnosis. The study period lasted from April 1st, 1978 until June 30th, 1986 and only individuals that were infected and diagnosed with AIDS during that period were included retrospectively. Data is subject to right truncation, because patients with long incubation intervals may not have developed AIDS at the time of enrollment (the end of the study), yet. Hence, events occurring after the time of enrollment cannot be recorded, which would result in underestimated survival probabilities if right truncation is ignored during inference.

In the examples above, truncation is incidental, but explicit inclusion or exclusion criteria of a study can lead to truncation too. For instance, a study interested in smoking among teenagers would need to explicitly define the age range that constitutes a teenager, thus estimates concerning the overall population would be biased.

## 2.3 Functions of Survival Time

In this section, I will define the basic quantities used in analyzing time-to-event data: 1) the probability density function of survival time, 2) the survival function, 3) the hazard function, and 4) the cumulative hazard function. In section 2.3.3, I will show that their definitions are actually mathematically equivalent, and therefore it suffices to define one of them.

### 2.3.1 Survival Function

> **Definition 2.1: Survival Function.** Let $T$ denote a continuous non-negative random variable corresponding to a patient's survival time. The survival function $S(t)$ returns the probability of survival beyond time $t$ and is defined as
>
> $$S(t) = P(T > t). \tag{2.1}$$

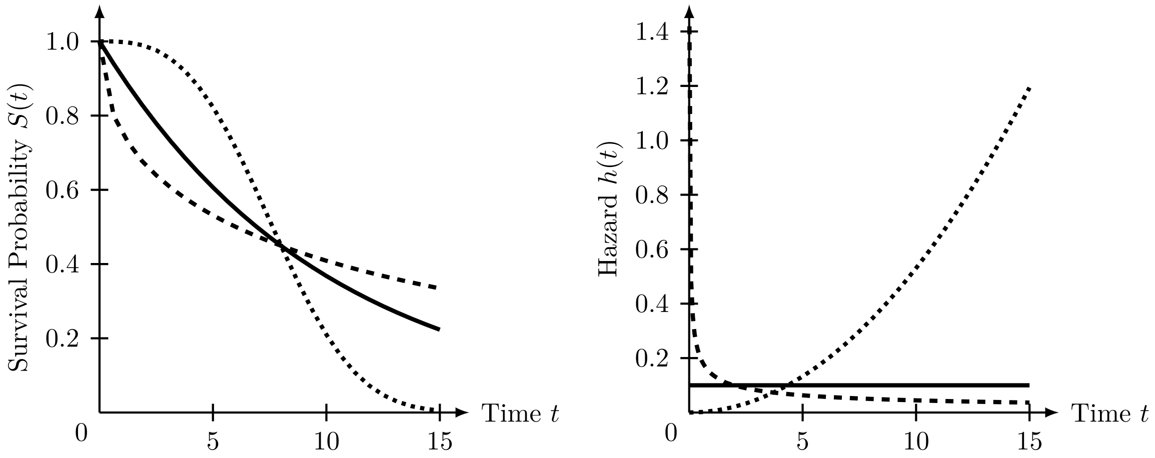


**Figure 2.2**: Survival and hazard functions following Weibull distribution: $S(t) = \exp(-(\lambda t)^k)$ and $h(t) = \lambda k(\lambda t)^{k-1}$ with $\lambda, k > 0$. For solid lines $\lambda = 0.1$ and $k = 1$, for dashed lines $\lambda = 0.08$ and $k = 0.5$, and for dotted lines $\lambda = 0.25 \cdot \sqrt[3]{0.1} \approx 0.1160$ and $k = 3$.

Moreover, $S(t)$ is non-increasing with $S(0) = 1$, and $S(\infty) = 0$. Alternatively, the survival function can be defined based on the cumulative distribution function $F(t)$ or the probability density function $f(t)$:

$$S(t) = P(T > t) = 1 - P(T \leq t) = 1 - F(t) = \int_t^\infty f(u)du. \tag{2.2}$$

**Example 2.3.** A popular choice for defining the survival function in a parametric form is the Weibull distribution with parameters $\lambda > 0$ and $k > 0$:

$$F(t) = 1 - \exp(-(\lambda t)^k) \tag{2.3}$$

$$S(t) = \exp(-(\lambda t)^k). \tag{2.4}$$

The plot in fig. 2.2 shows the survival function for three different configurations of $\lambda$ and $k$. If $k < 1$ (dashed line), the survival function is characterized by a high gradient at early times that decreases over time. Such a function could for example describe infant mortality, where the risk of death is the highest directly after birth and decreases over time. In contrast, if $k > 1$ (dotted line), the gradient of the survival function is relatively low at the beginning but increases over time. This formulation could be useful when modeling an aging process. If $k = 1$ (solid line), the survival function is between the aforementioned two curves and its interpretation will become obvious when discussing hazard functions in section 2.3.2.

Usually, the parametric form of the survival function is unknown and one has to resort to non-parametric estimators of the survival function from a given set of patients and their observed survival times. If data is uncensored, i.e., all individuals experienced an event before the study ended, the survival function at time $t$ can simply be estimated by the ratio of patients surviving beyond time $t$ and the total number of patients:

$$\hat{S}(t) = \frac{\text{number of patients surviving beyond } t}{\text{total number of patients}}. \tag{2.5}$$

In the presence of censoring, this estimator cannot be used, because the numerator is not always defined. For instance, consider the following set of patients, where $t_i$ and $\delta_i$ denote the survival time and event indicator of the $i$-th patient.

| $i$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $t_i$ | 4 | 6 | 6 | 7 | 9 |
| $\delta_i$ | 1 | 1 | 0 | 0 | 1 |

Using eq. (2.5), it is possible to compute $S(5) = \frac{4}{5} = 0.8$, but not $S(8)$, because it is unknown whether the third and fourth patient experienced an event before or after $t = 8$. An alternative non-parametric estimator that can be used for censored data is the Kaplan-Meier estimator described in section 2.4.

## 2.3.2 Hazard Function

In addition to the survival function, the hazard function is an important quantity in survival analysis, which I define next.

> **Definition 2.2: Hazard Function.** The hazard function $h(t)$ denotes an approximate probability (it is not bounded from above) that an event occurs in the small time interval $[t; t + \Delta t[$, under the condition that an individual would remain event-free up to time $t$:
>
> $$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t \mid T \geq t)}{\Delta t} \geq 0. \tag{2.6}$$

Alternative names for the hazard function are *conditional failure rate*, *conditional mortality rate*, or *instantaneous failure rate*. In contrast to the survival function, which describes the absence of an event, the hazard function provides information about the occurrence of an event.

**Example 2.4.** Continuing with example 2.3 from above, where the survival function followed a Weibull distribution, we can define the corresponding hazard function as $h(t) = \lambda k (\lambda t)^{k-1}$ (see fig. 2.2). The parameter $k$ of the Weibull distribution influences the form of the hazard function in the following ways. If $k = 1$ (solid line), the corresponding hazard function remains constant, meaning it is equally likely to

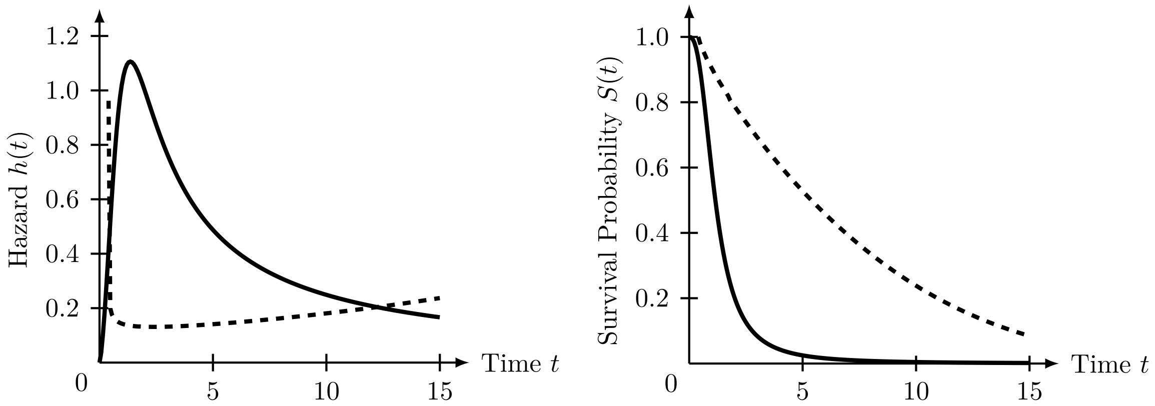


**Figure 2.3**: Examples of humpshaped (solid line) and bathtub shaped (dashed line) hazard functions. The humpshaped hazard function is based on a log-logistic distribution function and the bathtub shaped hazard function on an exponential power distribution.

experience an event at any point in time. If $k < 1$ (dashed line), the highest rate of experiencing an event occurs at early times and quickly decreases afterwards. Finally, if $k > 1$ (dotted line), the rate of events increases with time. Note that although survival functions in fig. 2.2 look relatively similar to each other, the corresponding hazard functions are dramatically different.

In addition to constant, increasing and decreasing hazard functions, humpshaped and bathtub shaped curves are commonly encountered as well (see fig. 2.3). The former can be used when investigating complications occurring after an intervention, where the risk of an event increases after surgery due to infection, bleeding, and so forth. A bathtub shaped hazard function is appropriate when modeling a population across the whole life span: early events are due to infant mortality and later events are due to the natural aging process.

If no record is censored, the hazard function at time $t$ can be estimated by the ratio of patients experiencing an event in the interval $[t; t + \Delta t[$ and the number of patients who remained event free up to time $t$:

$$\hat{h}(t) = \frac{\text{number of patients with event in time interval } [t; t + \Delta t[}{(\text{number of patients surviving up to time } t) \times \Delta t} \tag{2.7}$$

However, the estimate cannot be computed in the presence of censoring, because the exact time of an event is only known for a subset of patients.

Closely related to the hazard function is the cumulative hazard function, which is another basic quantity in survival analysis.

**Definition 2.3: Cumulative Hazard Function.** The cumulative hazard function $H(t)$ is the integral over the interval $[0; t]$ of the hazard function:

$$H(t) = \int_0^t h(u)du \tag{2.8}$$

The relationship between cumulative hazard function $H(t)$ and hazard function $h(t)$ is similar to the relationship between probability density function $f(t)$ and cumulative distribution function $F(t)$, except that $h(t)$ does not represent the density of a probability distribution.

### 2.3.3 Relationships between Functions

As mentioned in the beginning of this section, the definitions of survival function, hazard function and cumulative hazard function are all mathematically equivalent, and therefore it is sufficient to define one of them and derive the remaining two functions from it.

**Lemma 2.1.** *Let $T$ denote a continuous non-negative random variable, then the following relationships between probability density function $f(t)$, cumulative distribution function $F(t)$, survival function $S(t)$, hazard function $h(t)$, and cumulative hazard function $H(t)$ hold:*

$$f(t) = -\frac{d}{dt}S(t) = -S'(t) \tag{2.9}$$

$$h(t) = \frac{f(t)}{S(t)} \tag{2.10}$$

$$H(t) = -\log S(t) \Leftrightarrow S(t) = \exp(-H(t)) \tag{2.11}$$

*Proof.* The first equality is due to the definition $S(t) = 1 - F(t)$ in eq. (2.2), which when substituted on the right side yields $-(-\frac{d}{dt}F(t)) = f(t)$. The proof of the second equality is as follows:

$$\begin{aligned}
h(t) &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} P(t \leq T < t + \Delta t \mid T \geq t) \\
&= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \cdot \frac{P(\{t \leq T < t + \Delta t\} \cap \{T \geq t\})}{P(T \geq t)} \\
&= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \cdot \frac{P(t \leq T < t + \Delta t)}{P(T \geq t)} \\
&= \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} \cdot \frac{1}{1 - F(t)} \\
&= \frac{d}{dt}F(t) \cdot \frac{1}{1 - F(t)} = \frac{f(t)}{S(t)}.
\end{aligned}$$

Finally, using eq. (2.10) and substituting it in eq. (2.8), one obtains

$$\begin{aligned}H(t) &= \int_0^t h(u)du = \int_0^t \frac{f(u)}{S(u)}du = \int_0^t \frac{-S'(u)}{S(u)}du \\ &= -\left[\log S(u)\right]_0^t = -\log S(t) + \log S(0) \\ &= -\log S(t),\end{aligned}$$

with $\int \frac{f\prime(x)}{f(x)}dx = \log|f(x)| + b$ and $S(0) = 1$. □

**Example 2.5.** Using the results above, we can go back to example 2.3, where $f(t)$ follows a Weibull distribution with parameters $\lambda > 0$ and $k > 0$, to obtain the form of the survival function $S(t)$ and hazard function $h(t)$ as depicted in fig. 2.2. First of all, the probability density function and cumulative distribution function are defined as

$$f(t) = \lambda k(\lambda t)^{k-1}\exp(-(\lambda t)^k) \qquad \text{and} \qquad F(t) = 1 - \exp(-(\lambda t)^k).$$

Plugging these two quantities into the definition of $S(t)$, $h(t)$ and $H(t)$ results in

$$\begin{aligned}S(t) &= 1 - F(t) = 1 - (1 - \exp(-(\lambda t)^k)) = \exp(-(\lambda t)^k), \\ h(t) &= \frac{f(t)}{S(t)} = \frac{\lambda k(\lambda t)^{k-1}\exp(-(\lambda t)^k)}{\exp(-(\lambda t)^k))} = \lambda k(\lambda t)^{k-1}, \\ H(t) &= -\log S(t) = -\log(\exp(-(\lambda t)^k)) = (\lambda t)^k.\end{aligned}$$

### 2.3.4 Functions for Discrete Random Variables

Consider a study investigating the time to pregnancy. Because a woman can only get pregnant at certain time points in her menstrual cycle, the time to pregnancy would correspond to the number of menstrual cycles, which is discrete. The definition of survival function, hazard function and cumulative hazard function are slightly different in this case; most importantly, they are step functions.

**Definition 2.4: Discrete Survival Time.** Let $T$ be a discrete random variable, which can take on values $t_i$ ($i \in \mathbb{N}$) with probability mass function $P(T = t_i)$ and $t_i < t_j$ if and only if $i < j$, then

$$S(t) = \sum_{\{i|t_i > t\}} P(T = t_i) \tag{2.12}$$

$$P(T = t_i) = S(t_{i-1}) - S(t_i) \tag{2.13}$$

$$h(t_i) = P(T = t_i \mid T \geq t_i) \tag{2.14}$$

$$H(t) = \sum_{\{i|t_i \leq t\}} h(t_i), \tag{2.15}$$

where $S(t)$ is non-increasing with $S(t_0) = 1$.

From the definitions above, the connection between hazard and survival function is defined as

$$\begin{aligned} h(t_i) &= P(T = t_i \mid T \geq t_i) = \frac{P(T = t_i \cap T \geq t_i)}{P(T \geq t_i)} \\ &= \frac{P(T = t_i)}{P(T > t_{i-1})} = \frac{S(t_{i-1}) - S(t_i)}{S(t_{i-1})} \\ &= 1 - \frac{S(t_i)}{S(t_{i-1})}. \end{aligned} \tag{2.16}$$

Using $S(t_0) = S(0) = 1$, one can reformulate the survival function as

$$S(t_i) = \prod_{\{j \mid t_j \leq t_i\}} \frac{S(t_j)}{S(t_j)} S(t_i) = \frac{S(t_1)S(t_2) \cdots S(t_{i-1})S(t_i)}{S(t_0)S(t_1) \cdots S(t_{i-2})S(t_{i-1})} = \prod_{\{j \mid t_j \leq t_i\}} \frac{S(t_j)}{S(t_{j-1})} \tag{2.17}$$

to obtain

$$S(t) = \prod_{\{i \mid t_i \leq t\}} (1 - h(t_i)). \tag{2.18}$$

The biggest difference to $T$ being a continuous random variable is that the relationship $H(t) = -\log S(t)$ does not hold anymore, although the cumulative hazard function can be defined as $H(t) = -\sum_{\{i \mid t_i \leq t\}} \log(1 - h(t_i))$ to preserve this relationship [68].

## 2.4 Non-parametric Estimators

### 2.4.1 Estimators for Right Censored Survival Data

When collecting data in a clinical study, it is usually difficult to determine the exact parametric form of the distribution of survival time, which means that one has to resort to non-parametric estimators that derive the survival function $S(t)$ from a sample of right censored observations. In sections 2.3.1 and 2.3.2, I mentioned simple estimators of the survival function and hazard function that are only applicable if all patient records are uncensored. In contrast, the Kaplan-Meier estimator [167] of the survival function and the Nelson-Aalen estimator [2, 218] of the cumulative hazard function can be applied to censored data, i.e., when exact survival times are unknown for a subset of patients. Both estimators assume that the distribution of survival times is independent of the distribution of censoring times (non-informative censoring), such that knowing one of them does not provide additional information about the other.

Based on a sample $\mathcal{D} = \{(y_i, \delta_i)\}_{i=1}^n$ of $n$ patients, let $t_1 < t_2 < \cdots < t_m$ be the $m \leq n$ distinct time points where an event occurred ($\delta_i = 1$), and $d_1, d_2, \ldots, d_m$

the corresponding number of events at each of the $m$ time points. Moreover, let $\mathcal{R}_i = \{j | y_j \geq t_i\}$ denote the risk set, i.e., the set of patients who were still event-free shortly before time point $t_i$, then $\frac{d_i}{|\mathcal{R}_i|}$ is an estimate of the conditional probability $P(T = t_i \mid T \geq t_i)$ of experiencing an event at time point $t_i$, conditional on having remained event-free just prior to $t_i$. From this estimate, it is possible to define estimators of the survival and hazard function, as described below.

**Definition 2.5: Kaplan-Meier Estimator.** Under the assumption of non-informative censoring, the survival function $S(t)$ can be estimated from a sample of right censored survival data using the Kaplan-Meier estimator [167], which is defined as

$$\hat{S}(t) = \begin{cases} 1 & \text{if } t_1 > t, \\ \prod_{\{i|t_i \leq t\}} \left(1 - \frac{d_i}{|\mathcal{R}_i|}\right) & \text{if } t_1 \leq t, \end{cases} \tag{2.19}$$

where $i = 1, \ldots, m$.

**Definition 2.6: Nelson-Aalen Estimator.** Assuming non-informative censoring, the cumulative hazard function $H(t)$ can be estimated from a sample of right censored survival data using the Nelson-Aalen estimator [2, 218], defined as

$$\hat{H}(t) = \begin{cases} 0 & \text{if } t_1 > t, \\ \sum_{\{i|t_i \leq t\}} \frac{d_i}{|\mathcal{R}_i|} & \text{if } t_1 \leq t, \end{cases} \tag{2.20}$$

where $i = 1, \ldots, m$.

The Kaplan-Meier and Nelson-Aalen estimator are closely related to the definitions of survival and cumulative hazard function for discrete random variables when setting $\hat{h}(t_i) = \frac{d_i}{|\mathcal{R}_i|}$ (see section 2.3.4). Moreover, the value of both estimators is undefined if $t$ exceeds the largest observed time point ($t > \max_{i=1,\ldots,n} y_i$). Finally, the Kaplan-Meier estimator is also known as Product-Limit estimator.

**Example 2.6.** Figure 2.4 depicts estimated survival curves for 300 males and 200 females of the Worcester Heart Attack Study [146] using the Kaplan-Meier method. The figure shows that estimated survival functions are step functions having – by definition – an initial survival probability of one. Jumps in the estimated survival functions occur at time points where one or more events occurred. The example also shows that estimated survival functions are generally undefined beyond the largest observed time point. For the subgroup consisting of female patients, the largest observation corresponds to an event. Since a survival function is non-increasing, estimates beyond the largest observation would simply be zero, in this special case. In contrast, the largest observation in the male subgroup does not correspond to an event (it is censored) and therefore estimates beyond this time point are undefined.

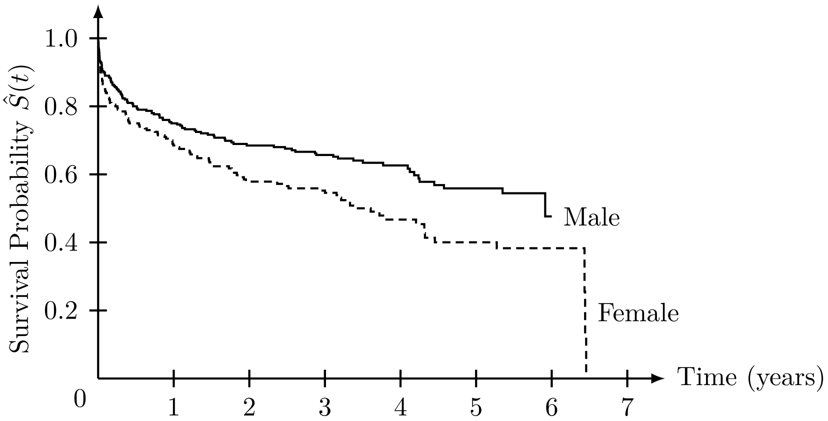


**Figure 2.4**: Estimated survival functions for males and females following hospital admission for acute myocardial infarction from data of the Worcester Heart Attack Study [146] using the Kaplan-Meier estimator.

## 2.4.2 Estimators for Truncated Survival Data

### Left Truncation

In the previous section, I discussed non-parametric estimators for right censored survival data. In this section, I will focus on data that is left truncated *and* right censored at the same time. Instead of estimating unconditional probabilities, estimators for left truncated data estimate conditional probabilities. Left truncated survival data consists of tuples $(l_i, y_i, \delta_i)$, where $l_i$ is the date the $i$-th subject entered the study, $y_i$ the time of an event or censoring, and $\delta_i$ a binary event indicator. The definition of the risk set $\mathcal{R}_i$ at time $t_i$ is the fundamental quantity for estimators for non-truncated right censored survival data in the previous section and needs to be redefined to be applicable to left truncated data. In addition to the event or censoring time, the risk set for left truncated data has to consider the time when an individual entered the study.

> **Definition 2.7.** The risk set $\mathcal{R}_i^{\text{left}}$ at time $t_i$ for left truncated and right censored survival data contains all observations who survived at least up to time $t_i$ and entered the study prior to time $t_i$:
>
> $$\mathcal{R}_i^{\text{left}} = \{j \mid y_j \geq t_i > l_j\}. \tag{2.21}$$

By replacing $\mathcal{R}_i$ with $\mathcal{R}_i^{\text{left}}$ in eqs. (2.19) and (2.20) one obtains conditional estimators of the survival function and cumulative hazard function, respectively [203]. The Kaplan-Meier estimator [167] becomes a *conditional* estimator of the survival function, because instead of estimating the unconditional probability $S(t) = P(T > t)$, it now estimates

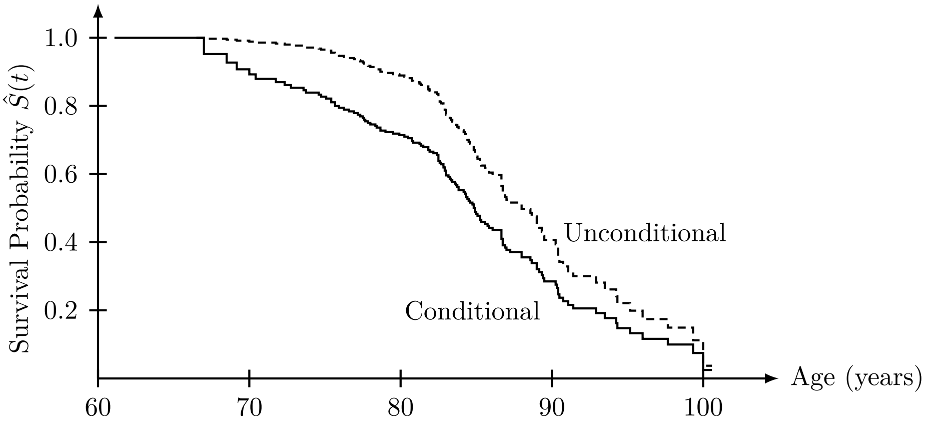


**Figure 2.5**: Estimated survival functions for 361 females of the Channing house retirement community [156] introduced in example 2.1. The solid line indicates the estimate conditional on survival at least up to 733 months (61.1 years), i.e., left truncation is considered. The dashed line indicates the unconditional estimate of the survival function that ignores left truncation.

the conditional probability

$$P(T > t \mid T \geq l_{\min}) = \frac{S(t)}{S(l_{\min})}, \tag{2.22}$$

with $l_{\min} = \min_{i=1,\ldots,n} l_i$ being the earliest time point a patient entered the study. Note that conditional estimates of the survival function for left truncated data have little meaning when the risk set $\mathcal{R}_i^{\text{left}}$ is small, which usually occurs for early time points (small values of $t_i$). In such cases, the Kaplan-Meier estimate should be limited to larger survival times:

$$\hat{S}_\alpha(t) = \prod_{\{i \mid \alpha \leq t_i \leq t\}} \left(1 - \frac{d_i}{|\mathcal{R}_i^{\text{left}}|}\right) \quad \text{if } t_1 \leq t, \tag{2.23}$$

where $0 < \alpha \leq t$.

**Example 2.7.** Figure 2.5 depicts conditional and unconditional estimates of the survival function using data of the Channing house retirement community that was introduced in example 2.1 on page 14. The unconditional estimate (dashed line) does not attribute the fact that people have to reach a certain age to enter the retirement community, therefore it ignores people that died earlier and overestimates survival probabilities compared to the conditional estimator (solid line). In contrast, the conditional survival curve considers the age of people when entering the community and thus is an appropriate estimator for left truncated and right censored survival data.

### Right Truncation

Estimation in the presence of right truncation, i.e., when individuals experiencing an event after truncation time $\nu$ are excluded from the study, follows a similar approach. Researchers are often confronted with right truncated data when studying infectious diseases in a retrospective study. The canonical example is AIDS, because subjects infected with HIV remain infected indefinitely and may eventually develop AIDS. In addition, individuals not infected with HIV will never get AIDS. Participants are enrolled in the study if they were diagnosed with the disease of interest in the chronological time interval $[0; \nu]$, otherwise a subject cannot be observed. The $i$-th individual is associated with the time of infection $u_i$ and the time $t_i$ between infection and diagnosis of the disease, called *lag*, which is of primary interest. Because data is right truncated, $u_i + t_i < \nu$ for all patients $(i = 1, \ldots, n)$.

Let $T$ and $U$ denote independent continuous non-negative random variables representing the lag and time of infection, respectively. The objective is to estimate the cumulative distribution function of the lag $T$, denoted by $F(t) = P(T \leq t)$. In general, $F(t)$ itself is unidentifiable, because subjects with $t > \nu$ cannot be observed. Instead, it is only possible to estimate the conditional cumulative distribution function

$$G(t) = \frac{F(t)}{F(t_{\max})} = P(T \leq t \mid T \leq t_{\max}), \tag{2.24}$$

where $t_{\max} = \max_{i=1,\ldots,n} t_i$ is the longest observed time between infection and onset. $G(t)$ can be estimated by transforming right truncated into left truncated data as proposed by Lagakos *et al.* [182].

Consider the reverse time $S = \nu - T$ and the relation $U + T \leq \nu$. The latter is satisfied if and only if $U \leq S$, which leads to the conclusion that it is only possible to observe subjects where $0 \leq U \leq S \leq \nu$, or equivalently that $S$ is left truncated by $U$.

> **Definition 2.8.** Let $\mathcal{R}_i^{\text{right}} = \{j \mid t_j \leq t_i \leq \nu - u_j\}$ denote the set of patients with lag smaller or equal to $t_i$ and time of infection smaller or equal to $s_i = \nu - t_i$, then the non-parametric estimator of $G(t)$ by Lagakos *et al.* [182] is given by
>
> $$\hat{G}(t) = \prod_{\{i \mid t \leq t_i \leq t_{\max}\}} \left(1 - \frac{d_i}{|\mathcal{R}_i^{\text{right}}|}\right). \tag{2.25}$$

For practical purposes, $\hat{G}(t)$ can be computed by applying the Kaplan-Meier estimator for left truncated data in eq. (2.22) to transformed observations with survival time

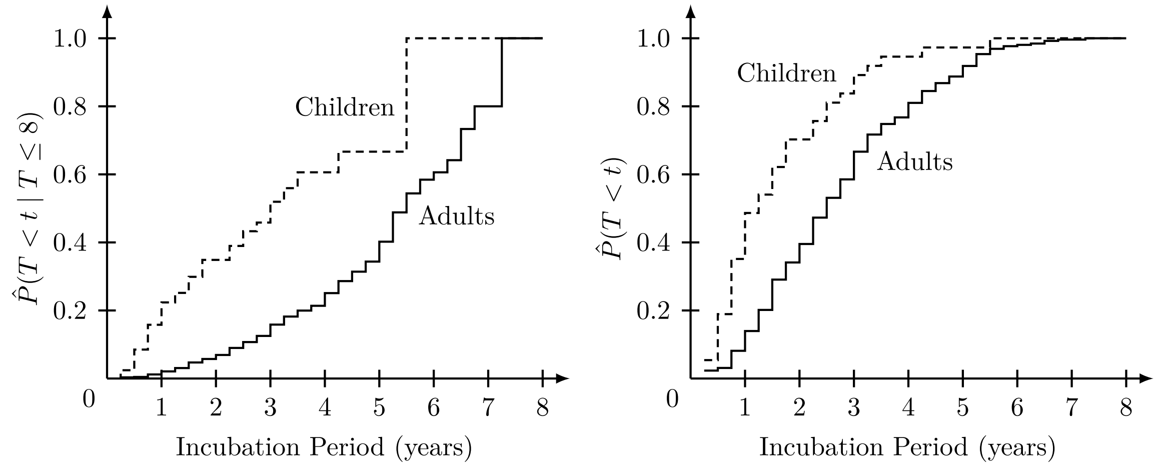


**Figure 2.6**: Non-parametric estimates of conditional cumulative distribution function $P(T < t \mid T \leq 8)$ and the unconditional cumulative distribution function $P(T < t)$. Data contains 258 adults and 37 children infected with HIV as described in example 2.2 [182]. Time was measured in three month intervals.

$t_i^* = -t_i$ and time of entry $l_i^* = -(\nu - u_i)$:

$$\begin{aligned}\hat{S}(-t) &= \prod_{\{i \mid -t_{\max} \leq t_i^* \leq -t\}} \left(1 - \frac{d_i}{|\{j \mid t_j^* \geq t_i^* \geq l_j^*\}|}\right) \\ &= \prod_{\{i \mid t \leq t_i \leq t_{\max}\}} \left(1 - \frac{d_i}{|\{j \mid t_j \leq t_i \leq \nu - u_j\}|}\right) = \hat{G}(t).\end{aligned}$$

Note, that the above estimator is only applicable if transformed observations are only left truncated, but not right censored ($\delta_i = 1$ for $i = 1, \ldots, n$).

**Example 2.8.** Returning to the AIDS dataset described in example 2.2 on page 14, the non-parametric estimator (2.25) can be used to estimate the conditional cumulative distribution function $P(T < t \mid T \leq \nu)$. The time was measured in years and discretized into three months intervals, therefore $\nu = 8$ denotes the end of study period. Figure 2.6 depicts estimates of $P(T < t \mid T \leq 8)$ for 258 adults and 37 children. Comparing the unconditional (right) to the conditional (left) estimate clearly shows that the rate of events was overestimated when ignoring right truncation.

## 2.5 Hypothesis Testing

When analyzing survival data, it is common for researchers to investigate whether survival times differ between groups of patients. In the simplest case, researchers are studying two groups of patients – such as treatment versus placebo or males versus females. A comparison of more than two groups usually arises if researchers want to determine how good a diagnostic model is in identifying low, medium and high risk patients. A first attempt might be the visual comparison of estimated survival functions of the respective patient groups. This provides a first impression on how well groups are separated with respect to survival, but does not provide a quantitative answer regarding the extent of differences or whether the differences are merely due to chance. Instead, it is preferred to perform statistical tests that systematically assess the differences between survival functions.

### 2.5.1 Two-Sample Log-Rank Test

When comparing two groups of patients with survival function $S_1(t)$ and $S_2(t)$, the objective is to compare the null hypothesis

$$H_0 : S_1(t) = S_2(t), \quad \forall 0 \leq t \leq \tau \quad \text{(patients of group 1 \& 2 have identical prognosis)}$$

against one of the alternatives

$$\begin{aligned} H_1 &: \exists 0 \leq t \leq \tau \mid S_1(t) > S_2(t) && \text{(group 1 has better prognosis than group 2)}, \\ H_2 &: \exists 0 \leq t \leq \tau \mid S_1(t) < S_2(t) && \text{(group 1 has worse prognosis than group 2)}, \\ H_3 &: \exists 0 \leq t \leq \tau \mid S_1(t) \neq S_2(t) && \text{(patients of group 1 \& 2 have different prognosis)}, \end{aligned}$$

where $\tau$ is the largest time point where all groups have at least one event-free subject. Because of censoring, traditional non-parametric tests such as the Wilcoxon test [317] or Mann-Whitney $U$ test [206] cannot be applied. Instead, special tests that account for censoring are preferred.

The *log-rank test* (or Mantel-Haenszel test) is the most commonly employed non-parametric test for comparing survival distributions [207, 208]. It can be applied under the assumption of non-informative censoring – survival times are independent of censoring times – and if survival curves do not cross. The test statistic is based on computing the difference between the observed number of events and the expected number of events at every distinct time point of an observed event. When considering two groups, the expected number of events can be obtained by multiplying the number of individuals at risk in group 1 by the proportion of the total number of individuals experiencing an event at time $t_i$ in both groups. Next, I will provide a formal definition of the two-sample log-rank test.

> **Definition 2.9: Two-Sample Log-Rank Test [207, 208].** Given a set of $m$ distinct time points of events, for group 1, let $d_{1i}$ denote the number of events at time $t_i$ and $r_{1i} = |\mathcal{R}_{1i}|$ the size of the risk set at time $t_i$, i.e., the set of patients that are still event-free shortly before time point $t_i$ ($i = 1, \ldots, m$). For group 2, $d_{2i}$ and $r_{2i}$ are defined similarly. The total number of events $d_i$, and the total number of individuals

at risk $r_i$ at time $t_i$ are defined as $d_i = d_{1i} + d_{2i}$ and $r_i = r_{1i} + r_{2i}$. Under the null hypothesis, $d_{1i}$ follows the hypergeometric distribution. Consequently, the expected number of events $e_{1i}$ in group 1 and its variance $\sigma_{1i}^2$ is given by

$$E(d_{1i}) = e_{1i} = r_{1i}\left(\frac{d_i}{r_i}\right) \tag{2.26}$$

$$\mathrm{Var}(d_{1i}) = \sigma_{1i}^2 = \frac{r_{1i}}{r_i} \cdot \frac{r_{2i}}{r_i}\left(\frac{r_i - d_i}{r_i - 1}\right) d_i. \tag{2.27}$$

The test statistic $X^2$ of the two-sample log-rank test is the difference between the overall number of observed events in group 1 ($d_{1i}$) and the overall number of expected events ($e_{1i}$):

$$X^2 = \frac{\left(\sum_{i=1}^{m} (d_{1i} - e_{1i})\right)^2}{\sum_{i=1}^{m} \sigma_{1i}^2}. \tag{2.28}$$

Under the null hypothesis, $X^2$ is approximately $\chi^2$-distributed with 1 degree of freedom. The null hypothesis should be rejected in favor of the one-sided alternative $S_1(t) < S_2(t)$ at significance level $\alpha$ if $X^2$ is larger than the upper $\alpha$ quantile of the $\chi^2$-distribution with 1 degree of freedom [176, p. 207].

### 2.5.2 Log-Rank Test for More Than Two Groups

If the survival distribution of more than two groups are to be compared, the log-rank test described above can be extended [233]. In this case, the objective is to determine whether the null hypothesis that patients in all groups have identical prognosis should be rejected in favor of the alternative that survival among some groups is significantly different:

$$H_0 : S_1(t) = S_2(t) = \cdots = S_K(t), \quad \forall 0 \leq t \leq \tau$$

versus

$$H_1 : \text{at least two of the survival functions } S_j(t) \text{ are not equal for some } 0 \leq t \leq \tau.$$

**Definition 2.10: Extended Log-Rank Test [233].** Given survival data for $K$ groups ($K \geq 2$), let $m$ denote total number of unique time points of events among all groups. The number of expected events at time $t_i$ in the $k$-th group can be defined similar to the two-sample log-rank test as

$$e_{ki} = r_{ki}\left(\frac{d_i}{r_i}\right), \tag{2.29}$$

where $r_{ki}$ denotes the number of patients of the $k$-th group at risk at time point $t_i$, $d_i = d_{1i} + \cdots + d_{Ki}$ the overall number of events at $t_i$, and $r_i = r_{1i} + \cdots + r_{Ki}$ the overall number of patients at risk. The test statistic is based on the following quantity of the $k$-th group

$$Z_k = \sum_{i=1}^{m} \left(d_{ki} - e_{ki}\right), \tag{2.30}$$

and its variance

$$\sigma_{kk}^2 = \sum_{i=1}^{m} \frac{r_{ki}}{r_i} \left(1 - \frac{r_{ki}}{r_i}\right) \left(\frac{r_i - d_i}{r_i - 1}\right) d_i \tag{2.31}$$

$$\sigma_{kg}^2 = -\sum_{i=1}^{m} \frac{r_{ki}}{r_i} \cdot \frac{r_{gi}}{r_i} \left(\frac{r_i - d_i}{r_i - 1}\right) d_i, \quad k \neq g, \tag{2.32}$$

where $k, g = 1, \dots, K$. Finally, the test statistic can be computed by selecting any $K-1$ of the $Z_k$'s and constructing the corresponding estimate of the $(K-1) \times (K-1)$ covariance matrix $\boldsymbol{\Sigma}$:

$$X^2 = \boldsymbol{Z}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{Z} \tag{2.33}$$

where $\boldsymbol{Z} = \left(Z_1(t), \cdots, Z_{K-1}(t)\right)^\top$. Under the null hypothesis, $X^2$ follows a $\chi^2$-distribution with $K-1$ degrees of freedom, therefore, the null hypothesis is rejected at significance level $\alpha$ if $X^2$ is larger than the upper $\alpha$ quantile of the $\chi^2$-distribution with $K-1$ degrees of freedom [176, p. 207].

### 2.5.3 Alternative Tests

The log-rank test gives equal weight to all distinct event times, therefore, it is most powerful when survival curves are parallel, i.e., the hazards are proportional. If researchers are more interested in detecting differences in survival distributions occurring early or late, the log-rank test can be extended by incorporating a non-negative, time-dependent weight function $w(t_i)$ into the test statistic.

**Definition 2.11.** The $k$-th element of the vector $\boldsymbol{Z}$ of the extendend log-rank test statistic (2.33) becomes

$$Z_k = \sum_{i=1}^{m} w(t_i) \left[d_{ki} - e_{ki}\right], \tag{2.34}$$

and entries in the covariance matrix

$$\sigma_{kk}^2 = \sum_{i=1}^{m} w(t_i)^2 \left[\frac{r_{ki}}{r_i} \left(1 - \frac{r_{ki}}{r_i}\right) \left(\frac{r_i - d_i}{r_i - 1}\right) d_i\right] \tag{2.35}$$

$$\sigma_{kg}^2 = -\sum_{i=1}^{m} w(t_i)^2 \left[\frac{r_{ki}}{r_i} \cdot \frac{r_{gi}}{r_i} \left(\frac{r_i - d_i}{r_i - 1}\right) d_i\right], \quad k \neq g. \tag{2.36}$$

Fleming and Harrington [102] proposed a general class of weight functions defined as

$$w_{\rho,\gamma}(t_i) = \hat{S}(t_i)^\rho (1 - \hat{S}(t_i))^\gamma, \quad \rho \geq 0, \gamma \geq 0, \tag{2.37}$$

**Table 2.1**: Weight functions used in various test statistics for comparing survival distributions.

| Test Statistic | $w(t_i)$ |
|---|---|
| Log-Rank [207] | $1$ |
| Gehan [108] | $r_i$ |
| Tarone and Ware [284] | $\sqrt{r_i}$ |
| Peto and Peto [233] | $\tilde{S}(t_i) = \prod_{\{j \mid t_j \leq t_i\}} (1 - d_j/(r_j + 1))$ |
| Fleming and Harrington [102] | $\hat{S}(t_i)^\rho(1 - \hat{S}(t_i))^\gamma$ |

where $\hat{S}$ is the Kaplan-Meier estimator (2.19) in the pooled sample. When setting $\rho = 1$ and $\gamma = 0$, the weight function emphasizes deviations at early times, whereas for $\rho = 0$ and $\gamma = 1$ late time points receive more weight. If $\rho$ and $\gamma$ are set to zero, all time points receive a constant weight as in the case of the log-rank test described above. Additional options for the weight function are summarized in table 2.1.

# 3 Predictive Models for Survival Analysis

The Kaplan-Meier and Nelson-Aalen estimator of the survival function and cumulative hazard function described in section 2.4 are non-parametric estimators that do not take into account patient characteristics when estimating survival curves. In this chapter, I will review survival models for right censored survival data that take into account a set of features or covariates to estimate a patient's risk of experiencing an adverse event. I will start by explaining classical models from statistics, namely the accelerated failure time model and Cox's proportional hazards model and continue by describing models that borrow ideas from machine learning, namely support vector machines, gradient boosting, and random forests.

I will show that most models build upon the set of basic concepts defined in chapter 2. The training data $\mathcal{D}$ for all models consists of $n$ patients, characterized by an observable time $y_i = \min(t_i, c_i) > 0$, a binary event indicator $\delta_i = I(t_i \leq c_i)$, and a $p$-dimensional feature vector $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^\top \in \mathbb{R}^p$:

$$\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^n.$$

## 3.1 Accelerated Failure Time Model

> **Definition 3.1: Accelerated Failure Time Model.** In the *accelerated failure time* (AFT) model (see e.g. [68, 176, 210, 269]), the survival function of an individual described by the feature vector $\boldsymbol{x} \in \mathbb{R}^p$ at time $t$ is equal to a baseline survival function $S_0$, evaluated at time $t \cdot \exp(-\boldsymbol{x}^\top \boldsymbol{\beta})$, i.e.,
>
> $$S(t|\boldsymbol{x}) = S_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta})), \tag{3.1}$$
>
> where $\boldsymbol{\beta} \in \mathbb{R}^p$ represents the model's coefficients.

From eq. (3.1) it is evident that features decelerate (if $\exp(-\boldsymbol{x}^\top \boldsymbol{\beta}) < 1$) or accelerate (if $\exp(-\boldsymbol{x}^\top \boldsymbol{\beta}) > 1$) the time to an event with respect to the baseline survival function.

**Lemma 3.1.** *The cumulative distribution function $F(t|\boldsymbol{x})$, the probability density function $f(t|\boldsymbol{x})$, and the hazard function $h(t|\boldsymbol{x})$ of the accelerated failure time model can be represented in form of the respective baseline functions (indicated by a subscript zero):*

$$
\begin{aligned}
F(t|\boldsymbol{x}) &= F_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta})), && (3.2)\\
f(t|\boldsymbol{x}) &= f_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta})) \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}), && (3.3)\\
h(t|\boldsymbol{x}) &= h_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta})) \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}). && (3.4)
\end{aligned}
$$

*Proof.* All equalities can be directly derived from definitions in eqs. (2.9) to (2.11) (see section 2.3.3 on page 18). The formulation for $F(t|\boldsymbol{x})$ can be obtained by substituting the definition of the survival function in eq. (2.2) into (3.2). Taking the derivative of (3.2) with respect to $t$ leads to the definition of $f(t|\boldsymbol{x})$:

$$
\begin{aligned}
\frac{d}{dt} F_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta})) &= \frac{d}{dt} F(t|\boldsymbol{x})\\
f_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta})) \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}) &= f(t|\boldsymbol{x}).
\end{aligned} \tag{3.5}
$$

Finally, starting with (3.1) and applying eq. (2.10) reveals the hazard function of the accelerated failure time model:

$$
\begin{aligned}
S(t|\boldsymbol{x}) &= S_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}))\\
\frac{f(t|\boldsymbol{x})}{h(t|\boldsymbol{x})} &= \frac{f_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}))}{h_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}))}\\
h(t|\boldsymbol{x}) &= \frac{f(t|\boldsymbol{x}) h_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}))}{f_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}))}\\
&= \frac{f(t|\boldsymbol{x}) h_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}))}{f(t|\boldsymbol{x}) \exp(\boldsymbol{x}^\top \boldsymbol{\beta})}\\
&= h_0(t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta})) \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}).
\end{aligned} \tag{3.6}
$$

□

Alternatively, the accelerated failure time model is often formulated as a linear model of the logarithm of the survival time:

$$
\begin{aligned}
\log t &= \beta_0 + \boldsymbol{x}^\top \boldsymbol{\beta} + \varepsilon\\
\Leftrightarrow t &= \exp(\beta_0 + \varepsilon) \exp(\boldsymbol{x}^\top \boldsymbol{\beta})\\
\Leftrightarrow t \exp(-\boldsymbol{x}^\top \boldsymbol{\beta}) &= \exp(\beta_0 + \varepsilon),
\end{aligned} \tag{3.7}
$$

where $\beta_0 \in \mathbb{R}$ is an intercept and $\boldsymbol{\beta}$ are the coefficients. The formulation as a linear model can be obtained from eq. (3.1) by letting $S_0(t)$ be the survival function of $\exp(\beta_0 + \varepsilon)$.

If the distribution of the baseline survival function, or equivalently the error term $\varepsilon$, remains unspecified, the model is called *semiparametric* accelerated failure time model, which I will focus on here. Given a set of $n$ patients, the semiparametric accelerated failure time model is defined as

$$\log t_i = \beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad \forall i = 1, \ldots, n, \tag{3.8}$$

where the error terms $\varepsilon_i$ are independent and identically distributed random variables. A naive approach would be to obtain estimates of the coefficients via ordinary least squares, however, due to right censoring it is only possible to observe $y_i = \min(t_i, c_i)$, therefore the ordinary least squares solution is only unbiased if all individuals are uncensored ($\delta_i = 1 \ \forall i = 1, \ldots, n$). Next, I will describe two methods to estimate $\beta_0$ and $\boldsymbol{\beta}$ in the accelerated failure time model: the Buckley-James estimator, and the inverse probability of censoring weighted least squares estimator.

### 3.1.1 Buckley-James Estimator

For the remainder of this section, I will use $t_i$, $c_i$, and $y_i$ to denote the logarithmic transformation of the survival time, censoring time, and observed time, respectively. Buckley and James [41] replaced the log survival time $t_i$ of censored observations in eq. (3.8) by $E(t_i | t_i > c_i, \boldsymbol{x}_i)$, which is equivalent to imputing survival times of censored records from the conditional expectation given an individual's time of censoring and feature vector. Consequently, the linear model in eq. (3.8) distinguishes between uncensored and censored samples:

$$t_i = \delta_i y_i + (1 - \delta_i) E(t_i | t_i > y_i, \boldsymbol{x}_i). \tag{3.9}$$

Note that for censored records $y_i = c_i$, because only the time of censoring $c_i$ can be observed. The conditional expectation has the form

$$\begin{aligned} E(t_i | t_i > c_i, \boldsymbol{x}_i) &= \beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta} + E(\varepsilon_i | \varepsilon_i > c_i - \beta_0 - \boldsymbol{x}_i^\top \boldsymbol{\beta}) \\ &= \beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta} + \int_{c_i - \beta_0 - \boldsymbol{x}_i^\top \beta}^{\infty} \left( \frac{u}{1 - F(c_i - \beta_0 - \boldsymbol{x}_i^\top \boldsymbol{\beta})} \right) dF(u) \end{aligned} \tag{3.10}$$

where the function $F$ is the cumulative distribution function of $T - \beta_0 - \boldsymbol{x}_i^\top \boldsymbol{\beta}$. The function $F(t)$ can be estimated by the Kaplan-Meier estimator (2.19) based on data $\{t_i - \beta_0 - \boldsymbol{x}_i^\top \boldsymbol{\beta}, \delta_i\}_{i=1}^n$ as described below.

> **Definition 3.2: Buckley-James estimator [41].** Let $r_i = y_i - \hat{\beta}_0 - \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}$ denote the residual of the $i$-th sample with respect to the current estimates of $\beta_0$ and $\boldsymbol{\beta}$, and let $\hat{S}(r_i)$ denote the Kaplan-Meier estimator of $1 - F(r_i)$ for the $i$-th residual, and $\Delta S(r_i)$ the step size of the Kaplan-Meier estimator at $r_i$. By using Kaplan-Meier estimates in place of $1 - F(\cdot)$ in (3.10), the survival time of censored subjects can be

imputed by

$$\tilde{t}_i = \hat{\beta}_0 + \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}} + \sum_{k=1}^{n} \delta_k w_{ik} \left( y_k - \hat{\beta}_0 - \boldsymbol{x}_k^\top \hat{\boldsymbol{\beta}} \right), \tag{3.11}$$

with

$$w_{ik} = \begin{cases} \Delta \hat{S}(r_k) [\hat{S}(r_i)]^{-1} & \text{if } r_i < r_k, \\ 0 & \text{else.} \end{cases} \tag{3.12}$$

Substituting eq. (3.11) into eq. (3.9), the survival time $\tilde{t}_i$ used in least-squares regression for right-censored data following Buckley and James [41] is

$$\tilde{t}_i = \delta_i t_i + (1 - \delta_i) \left[ \hat{\beta}_0 + \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}} + \sum_{k=1}^{n} \delta_k w_{ik} \left( c_k - \hat{\beta}_0 - \boldsymbol{x}_k^\top \hat{\boldsymbol{\beta}} \right) \right]. \tag{3.13}$$

Once the survival times of censored records have been imputed, an obvious choice would be to use ordinary least squares with the imputed outcomes to obtain estimates of the coefficients. Assuming both the features and the response are centered, i.e., $n^{-1} \sum_{i=1}^{n} \boldsymbol{x}_i = \boldsymbol{0}_p$ and $n^{-1} \sum_{i=1}^{n} \tilde{t}_i = 0$, the ordinary least squares estimates of $\beta_0$ and $\boldsymbol{\beta}$ are the solution to

$$- \boldsymbol{X}^\top (\tilde{\boldsymbol{t}} - \beta_0 - \boldsymbol{X}\boldsymbol{\beta}) = 0, \tag{3.14}$$

with $\tilde{\boldsymbol{t}} = (\tilde{t}_1, \ldots, \tilde{t}_n)^\top$. However, imputed survival times $\tilde{t}_i$ depend on the unknown coefficients $\beta_0$ and $\boldsymbol{\beta}$, which makes eq. (3.14) neither continuous nor monotone in $\beta_0$ and $\boldsymbol{\beta}$ [163].

Jin *et al.* [163] employed an iterative expectation maximization algorithm that first imputes survival times given an initial estimate of the coefficients, and then updates the coefficients by the ordinary least squares solution as

$$\hat{\boldsymbol{\beta}}^{\text{new}} = \left( \boldsymbol{X}^\top \boldsymbol{X} \right)^{-1} \boldsymbol{X}^\top \tilde{\boldsymbol{t}}, \tag{3.15}$$

$$\hat{\beta}_0^{\text{new}} = \frac{1}{n} \sum_{i=1}^{n} \tilde{t}_i - \boldsymbol{X} \hat{\boldsymbol{\beta}}^{\text{new}}. \tag{3.16}$$

They proposed to use a Gehan-type estimator [108] to obtain the initial estimate of the coefficients by solving the convex optimization problem

$$\min_{\beta_0, \boldsymbol{\beta}} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_i \max(0, r_j - r_i) = \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_i \max(0, y_j - \boldsymbol{x}_j^\top \boldsymbol{\beta} - y_i + \boldsymbol{x}_i^\top \boldsymbol{\beta}). \tag{3.17}$$

### 3.1.2 Inverse Probability of Censoring Weighted Least Squares

Stute [281] proposed an alternative to the Buckley-James estimator described in the previous section. Let the censoring mechanism be described by the conditional censoring survivor function $G(c \mid \boldsymbol{x}) = P(C > c \mid \boldsymbol{x})$. Instead of imputing survival times of censored subjects, he assigned each sample a weight $\omega_i$ proportional to the inverse probability of being censored after time $y_i$, given the feature vector $\boldsymbol{x}_i$:

$$\omega_i = \frac{\delta_i}{\hat{G}(y_i \mid \boldsymbol{x}_i)}, \tag{3.18}$$

where $\hat{G}(\cdot)$ is an estimator of the conditional censoring survivor function and random censoring is assumed to ensure $G(\cdot) > 0$. By considering sample weights $\omega_i$ and applying a logarithmic transformation to the observed time points, the objective becomes

$$\operatorname*{argmin}_{\beta_0, \boldsymbol{\beta}} \quad \frac{1}{n} \sum_{i=1}^{n} \omega_i (\log y_i - \beta_0 - \boldsymbol{x}_i^\top \boldsymbol{\beta}). \tag{3.19}$$

Due to the form of the weights in eq. (3.18), the algorithm is referred to as *inverse probability of censoring (IPC) weighted least squares* [49, 240, 301].

*Note.* By assumption, $G(Y \mid \boldsymbol{x}) \geq G(T \mid \boldsymbol{x}) > 0$ holds; hence, the weights $\omega_i$ always exist. In situations where the assumption of random censoring does not hold, $\hat{G}(c \mid \boldsymbol{x})$ is not guaranteed to be strictly positive. For instance, this could be the case for type I censoring, i.e., when the time of censoring is fixed in advance, because events after the end of the study could never be observed. In addition, it is often assumed, for practical purposes, that censoring is independent of the features, i.e., $G(c \mid \boldsymbol{x}) = G(c)$, which allows using the non-parametric Kaplan-Meier estimator (2.19) based on data $\mathcal{D} = \{(y_i, 1 - \delta_i)\}_{i=1}^n$ to estimate $\hat{G}(y_i \mid \boldsymbol{x}_i)$ in eq. (3.18).

### Penalized AFT Models

Finally, I want to mention that both the Buckley-James estimator as well as IPC weighted least squares have been extended to include a Least Absolute Shrinkage and Selection Operator (LASSO; [288]) or elastic net penalty [332], which makes the AFT model suitable for high-dimensional data. An AFT model with LASSO penalty based on IPC weights was proposed in [152], and based on the Buckley-James estimator in [164]. Extensions to the elastic net penalty have been discussed in [87, 308]. For a more detailed discussion on penalized models see section 3.2.5 (page 43).

## 3.2 Cox's Proportional Hazards Model

Cox's proportional hazards model [67] is a linear semiparametric model for right censored survival data. It is by far the most cited predictive survival model [303], because it is a fundamental tool in clinical research to identify risk factors of a particular disease. Several extensions to the Cox model have been proposed to adapt it to data with multicolinearities and high-dimensional feature vectors.

### 3.2.1 The Proportional Hazards Model

> **Definition 3.3: Cox's Proportional Hazards Model.** Cox's proportional hazards model [67] models the hazard function of the $i$-th patient, conditional on the feature vector $\boldsymbol{x}_i \in \mathbb{R}^p$, as the product of an unspecified baseline hazard function $h_0$ and an exponential function of the linear model $\boldsymbol{x}_i^\top \boldsymbol{\beta}$:
>
> $$h(t|x_{i1}, \ldots, x_{ip}) = h_0(t) \exp\left( \sum_{j=1}^{p} x_{ij}\beta_j \right) \Leftrightarrow \log \frac{h(t|\boldsymbol{x}_i)}{h_0(t)} = \boldsymbol{x}_i^\top \boldsymbol{\beta}, \tag{3.20}$$
>
> where $\boldsymbol{\beta} \in \mathbb{R}^p$ are the coefficients associated with each of the $p$ features, and no intercept term is included in the model.

The baseline hazard function $h_0$ only depends on the time $t$, whereas the exponential is independent of time and only depends on the covariates $\boldsymbol{x}_i$. Thus, $h_0$ is used to represent changes in risk over time and is the hazard function if one would ignore all features. An alternative interpretation of Cox's proportional hazards model [67] is that the linear model $\boldsymbol{x}_i^\top \boldsymbol{\beta}$ denotes the log ratio of the $i$-th individual's hazard function to the baseline hazard function (see the right hand side of eq. (3.20)).

The ratio of the hazard functions of two individuals, the so-called *hazard ratio*, can be interpreted similar to the odds ratio in logistic regression. For any two feature vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, the hazard ratio is given by

$$\frac{h(t|\boldsymbol{x}_1)}{h(t|\boldsymbol{x}_2)} = \frac{h_0(t)\exp(\boldsymbol{x}_1^\top \boldsymbol{\beta})}{h_0(t)\exp(\boldsymbol{x}_2^\top \boldsymbol{\beta})} = \exp(\boldsymbol{x}_1^\top \boldsymbol{\beta} - \boldsymbol{x}_2^\top \boldsymbol{\beta}) = \exp\left( \sum_{j=1}^{p} (x_{1j} - x_{2j})\beta_j \right). \tag{3.21}$$

If all features are fixed and only the $j$-th feature is incremented by 1, the hazard ratio simplifies to

$$\frac{h(t|x_1, \ldots, x_j, \ldots, x_p)}{h(t|x_1, \ldots, x_j + 1, \ldots, x_p)} = \exp(\beta_j). \tag{3.22}$$

In the general case in eq. (3.21) as well as the specific case in eq. (3.22), the hazard ratio is a constant independent of time, which is referred to as the *proportional hazards assumption*. In other words, the proportional hazards assumption means that the ratio of the "risk" (hazard) of experiencing an event of two subjects is constant over time. Hence, Cox's proportional hazards model [67] can only accurately estimate survival if the proportional hazards assumption holds.

**Example 3.1.** Once the coefficients $\boldsymbol{\beta}$ have been estimated from training data $\mathcal{D}$, their values provide information about their effect on the hazard. For illustration, assume that there is only a single binary feature that denotes whether a patient was treated with a newly developed drug ($x_{i1} = 1$) or received a placebo ($x_{i1} = 0$). When comparing a patient receiving the new drug to patient receiving the placebo, the hazard ratio becomes

$$\frac{h(t|x_{11}=1)}{h(t|x_{21}=0)} = \exp(\beta_1) \Leftrightarrow \log\left(\frac{h(t|x_{11}=1)}{h(t|x_{21}=0)}\right) = \beta_1.$$

If the new drug is ineffective, $h(t|x_{11}=1) = h(t|x_{21}=0)$ and the coefficient $\beta_1$ would be zero. If the new drug is protective of the disease, $h(t|x_{11}=1) < h(t|x_{21}=0)$ and $\beta_1 < 0$. Finally, if the new drug is harmful, $h(t|x_{11}=1) > h(t|x_{21}=0)$ and $\beta_1 > 0$. Therefore, the sign of the coefficient indicates whether the new drug has a negative or positive impact on survival.

### 3.2.2 Model Fitting

Fitting Cox's proportional hazards model (3.20) is achieved by maximizing a partial likelihood function with respect to the coefficients $\boldsymbol{\beta}$ using the Newton-Raphson algorithm. I will first derive the full likelihood function, followed by describing the partial likelihood function and the details of its maximization.

#### Full Likelihood Function

The main assumption for all likelihood-based estimators from survival data is that survival time and censoring time are independent, which is a requirement for non-parametric estimators described in section 2.4 as well. Assuming samples are independently and identically distributed, the general form of the likelihood function for $m$ distinct time points is defined as

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{m} P(T = t_i),$$

where $P(T = t_i)$ is the probability of an event at time point $t_i$.

In practice, the exact time point $t_i$ of an event is only known for a subset of patients, whereas for the remaining patients only the time of censoring $c_i$ is known. As before, let $T$ and $C$ denote a non-negative random variable representing the survival time and censoring time, respectively, then only $Y = \min(T, C)$ can be observed. When constructing the likelihood, it is necessary to consider that the time of an event is only partially known for censored records, whereas the exact time of an event is known for uncensored records. Thus, the training data $\mathcal{D}$ can be decomposed into two disjoint sets $\mathcal{D} = \mathcal{T} \cup \mathcal{C}$, where $\mathcal{T}$ contains all subjects that experienced an event, and $\mathcal{C}$ contains all subjects that are right censored, which yields the likelihood function

$$L(\boldsymbol{\beta}) = \prod_{i \in \mathcal{T}} P(Y = t_i \cap \delta_i = 1) \prod_{i \in \mathcal{C}} P(Y = c_i \cap \delta_i = 0).$$

A censored record only provides information about the probability $P(T > c_i)$, whereas an uncensored record provides information with respect to $P(T = t_i)$. For $i \in \mathcal{T}$,

$$\begin{aligned} P(Y = t_i \cap \delta_i = 1) &= P(Y = t_i \mid \delta_i = 1)P(\delta_i = 1) \\ &= P(T = t_i \mid T \leq c_i)P(T \leq c_i)) \\ &= \frac{f(t_i)}{F(t_i)} F(t_i) = f(t_i) \end{aligned} \tag{3.23}$$

and for $i \in \mathcal{C}$,

$$P(Y = t_i \cap \delta_i = 0) = P(T > c_i) = S(c_i). \tag{3.24}$$

Consequently, the likelihood function for right censored survival data can be compactly expressed as

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{m} [f(y_i)]^{\delta_i} [S(y_i)]^{1-\delta_i} \tag{3.25}$$

Substituting $f(t) = h(t)S(t)$ and $S(t) = \exp(-H(t))$ from eqs. (2.10) and (2.11) into eq. (3.25) yields an alternative definition of the full likelihood function of the form

$$\begin{aligned} L(\boldsymbol{\beta}) &= \prod_{i=1}^{m} [h(y_i)S(y_i)]^{\delta_i} [\exp(-H(y_i))]^{1-\delta_i} \\ &= \prod_{i=1}^{m} [h(y_i)\exp(-H(y_i))]^{\delta_i} [\exp(-H(y_i))]^{1-\delta_i} \\ &= \prod_{i=1}^{m} [h(y_i)]^{\delta_i} \exp(-H(y_i)). \end{aligned} \tag{3.26}$$

### Partial Likelihood Function Without Tied Survival Times

When substituting the definition of the hazard function $h(t)$ and the cumulative hazard function $H(t)$ of Cox's proportional hazards model (3.20) into eq. (3.26), optimization has be to performed with respect to $\boldsymbol{\beta}$ and the unknown baseline hazard function $h_0(t)$. This is generally not possible and led to the proposition of Cox [67] to only include conditional probabilities in the likelihood function. I will first present Cox's original formulation, which assumes that there are no ties in survival times, and subsequently describe suitable estimators for data with tied survival times.

The quantity of interest is the probability that the $i$-th individual experiences an event at time $t_i$, given that there is one event at time point $t_i$. This conditional probability can be defined in terms of the exponential function in eq. (3.20) such that the baseline hazard function can be eliminated from the likelihood function [176, p. 257]:

$$\begin{aligned}
&P(\text{subject experiences event at } y_i \mid \text{one event at } y_i) \\
=&\frac{P(\text{subject experiences event at } y_i \mid \text{event-free up to } y_i)}{P(\text{one event at } y_i \mid \text{event-free up to } y_i)} \\
=&\frac{h(y_i|\boldsymbol{x}_i)}{\sum_{j=1}^{n} I(y_j \geq y_i) h(y_j|\boldsymbol{x}_j)} = \frac{h_0(y_i)\exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})}{\sum_{j=1}^{n} I(y_j \geq y_i) h_0(y_j)\exp(\boldsymbol{x}_j^\top \boldsymbol{\beta})} \\
=&\frac{\exp(\boldsymbol{x}_i^\top \beta)}{\sum_{j=1}^{n} I(y_j \geq y_i)\exp(\boldsymbol{x}_j^\top \beta)} = \frac{\exp(\boldsymbol{x}_i^\top \beta)}{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \beta)},
\end{aligned} \tag{3.27}$$

where $I(\cdot)$ is the indicator function and $\mathcal{R}_i = \{j \mid y_j \geq y_i\}$ is the risk set, i.e., the set of patients who remained event-free shortly before time point $y_i$. By multiplying the conditional likelihood from above for all patients who experienced an event, Cox [67] constructed the *partial likelihood function*

$$PL(\boldsymbol{\beta}) = \prod_{i=1}^{n} \left[ \frac{\exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})}{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta})} \right]^{\delta_i}. \tag{3.28}$$

### Optimization

Instead of maximizing the partial likelihood function, it is numerically more stable to maximize the log partial likelihood function instead:

$$\log PL(\boldsymbol{\beta}) = \sum_{i=1}^{n} \delta_i \left[ \boldsymbol{x}_i^\top \boldsymbol{\beta} - \log \left( \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) \right) \right]. \tag{3.29}$$

The log partial likelihood function is convex in $\boldsymbol{\beta}$, hence the Newton-Raphson algorithm can be employed to obtain estimates $\hat{\boldsymbol{\beta}}$ of the coefficients from training data (see algorithm 3.1).

Before deriving the gradient and Hessian of the log partial likelihood function, additional notations are required. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top$ denote the $n \times p$ matrix of all feature vectors, and $\mathbb{1}_{\mathcal{R}_i} \in \{0,1\}^n$ the indicator vector of the risk set, such that the $j$-th element of $\mathbb{1}_{\mathcal{R}_i}$ is 1 if $y_j \geq y_i$ and zero otherwise. Moreover, $\bar{\boldsymbol{X}}_i = \operatorname{diag}(\mathbb{1}_{\mathcal{R}_i})\boldsymbol{X}$ is a modified version of $\boldsymbol{X}$, where rows corresponding to individuals that are not in the risk set $\mathcal{R}_i$ are set to zero. Finally, let $w_i \in \mathbb{R}$ be the denominator of the conditional probability (3.27): $w_i = \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) = \mathbb{1}_{\mathcal{R}_i}^\top \exp(\boldsymbol{X}\boldsymbol{\beta})$.

The first-order derivative of the log partial likelihood function (3.29) is defined as

$$\begin{aligned}
\frac{\partial \log PL(\boldsymbol{\beta})}{\partial \beta_k} &= \sum_{i=1}^{n} \delta_i \left[ x_{ik} - \frac{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk}}{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta})} \right] \\
&= \sum_{i=1}^{n} \delta_i \left[ x_{ik} - \frac{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk}}{w_i} \right],
\end{aligned} \tag{3.30}$$

which can be written in matrix form as

$$\frac{\partial \log PL(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \delta_i \left[ \boldsymbol{x}_i - \frac{1}{w_i} \bar{\boldsymbol{X}}_i^\top \operatorname{diag}(\exp(\boldsymbol{X}\boldsymbol{\beta})) \mathbb{1}_n \right]. \tag{3.31}$$

The diagonal elements of the Hessian matrix are defined as

$$\begin{aligned} \frac{\partial^2 \log PL(\boldsymbol{\beta})}{\partial \beta_k^2} &= -\sum_{i=1}^{n} \frac{\delta_i}{w_i^2} \left[ \left( \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk}^2 \right) w_i - \left( \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk} \right)^2 \right] \\ &= -\sum_{i=1}^{n} \delta_i \left[ \frac{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk}^2}{w_i} - \left( \frac{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk}}{w_i} \right)^2 \right], \end{aligned} \tag{3.32}$$

and the off-diagonal elements as

$$\begin{aligned} \frac{\partial^2 \log PL(\boldsymbol{\beta})}{\partial \beta_k \partial \beta_v} &= -\sum_{i=1}^{n} \frac{\delta_i}{w_i^2} \left( \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk} x_{jv} \right) w_i \\ &\quad - \frac{\delta_i}{w_i^2} \left( \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk} \right) \left( \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jv} \right) \\ &= -\sum_{i=1}^{n} \delta_i \frac{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk} x_{jv}}{w_i} \\ &\quad - \delta_i \left( \frac{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk}}{w_i} \right) \left( \frac{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jv}}{w_i} \right) \\ &= -\sum_{i=1}^{n} \frac{\delta_i}{w_i^2} \left[ w_i \left( \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk} x_{jv} \right) - \right. \\ &\quad \left. \left( \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jk} \right) \left( \sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) x_{jv} \right) \right] \end{aligned} \tag{3.33}$$

In matrix form, the second-order derivative becomes

$$\frac{\partial^2 \log PL(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = \sum_{i=1}^{n} \frac{\delta_i}{w_i^2} \bar{\boldsymbol{X}}_i^\top \left[ w_i \cdot \operatorname{diag}(\exp(\boldsymbol{X}\boldsymbol{\beta})) - \exp(\boldsymbol{X}\boldsymbol{\beta}) \exp(\boldsymbol{X}\boldsymbol{\beta})^\top \right] \bar{\boldsymbol{X}}_i. \tag{3.34}$$

### Partial Likelihood Function When Ties Are Present

The original formulation of the partial likelihood function by Cox [67] did not consider ties in survival times. In practice however, ties in survival times are common because survival times cannot be recorded with arbitrary precision; in most clinical trials, survival times are recorded in days, thus the survival times of people experiencing

**Algorithm 3.1:** Maximization of log partial likelihood function (3.29) of Cox's proportional hazards model using Newton-Raphson algorithm.

```
Input: Training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^n$.
Output: Estimate of coefficients $\hat{\boldsymbol{\beta}}$.
$\boldsymbol{\beta}^0 \leftarrow \mathbf{0}_p$
$t \leftarrow 0$
while not converged do
    Compute Newton step
        $\Delta\boldsymbol{\beta} = \left( \frac{\partial^2 - \log PL(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\top} \right)^{-1} \frac{\partial - \log PL(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}$
    with $\boldsymbol{\beta} = \boldsymbol{\beta}^t$.
    Update $\boldsymbol{\beta}^{t+1} \leftarrow \boldsymbol{\beta}^t - \Delta\boldsymbol{\beta}$
    $t \leftarrow t + 1$
end
$\hat{\boldsymbol{\beta}} \leftarrow \boldsymbol{\beta}^t$
```

an event on the same day are indistinguishable. In this case, the partial likelihood function has to consider all possible ways patients with identical survival times can be ordered, which amounts to $m!$ ($m$ factorial) possible permutations [73]. Hence, computing the exact partial likelihood in the presence of tied survival times quickly becomes intractable.

Breslow [40] proposed a straightforward extension of the partial likelihood function (3.28) by simply treating tied survival times as distinct:

$$PL_{\text{Breslow}}(\boldsymbol{\beta}) = \prod_{i=1}^{m} \left[ \frac{\exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})}{\left[\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta})\right]^{d_i}} \right]^{\delta_i}, \tag{3.35}$$

where $m$ is the total number of distinct time points and $d_i$ the number of events at time point $t_i$.

An alternative formulation that is closer to the exact partial likelihood when many ties are present was proposed by Efron [81]:

$$PL_{\text{Efron}}(\boldsymbol{\beta}) = \prod_{i=1}^{m} \left[ \frac{\exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})}{\prod_{k=1}^{d_i} \left[\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta}) - \frac{k-1}{d_i} \sum_{j \in \mathcal{D}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta})\right]} \right]^{\delta_i}, \tag{3.36}$$

where $\mathcal{D}_i = \{j \mid t_j = t_i \wedge \delta_j = 1\}$ is the set of patients who have experienced an event at time point $t_i$, and $d_i = |\mathcal{D}_i|$ its cardinality.

### 3.2.3 Estimation of Survival Function

Once the coefficients $\boldsymbol{\beta}$ of Cox's proportional hazards model have been estimated, it is often of interest to estimate the survival function of new patients, based on the fitted model. Estimation of an individual's survival function with feature vector $\boldsymbol{x}_{\text{new}} \in \mathbb{R}^p$ is based on the definition of the survival function in eq. (2.11), i.e., $S(t \mid \boldsymbol{x}_{\text{new}}) = \exp(-H(t \mid \boldsymbol{x}_{\text{new}}))$.

Starting from the definition of the cumulative hazard function (2.8) and substituting the hazard function (3.20), the cumulative hazard function for Cox's proportional hazards model can be expressed as

$$\begin{aligned} H(t \mid \boldsymbol{x}_{\text{new}}) &= \int_0^t h(u \mid \boldsymbol{x}_{\text{new}}) du \\ &= \int_0^t h_0(u) \exp(\boldsymbol{x}_{\text{new}}^\top \boldsymbol{\beta}) du \\ &= \exp(\boldsymbol{x}_{\text{new}}^\top \boldsymbol{\beta}) \int_0^t h_0(u) du \\ &= \exp(\boldsymbol{x}_{\text{new}}^\top \boldsymbol{\beta}) H_0(t). \end{aligned} \tag{3.37}$$

By re-substituting eq. (3.37) into eq. (2.11), the survival function for Cox's proportional hazards model becomes

$$\begin{aligned} S(t \mid \boldsymbol{x}_{\text{new}}) &= \exp(-H(t \mid \boldsymbol{x}_{\text{new}})) \\ &= \exp(-\exp(\boldsymbol{x}_{\text{new}}^\top \boldsymbol{\beta}) H_0(t)) \\ &= [\exp(-H_0(t))]^{\exp(\boldsymbol{x}_{\text{new}}^\top \boldsymbol{\beta})} \\ &= S_0(t)^{\exp(\boldsymbol{x}_{\text{new}}^\top \boldsymbol{\beta})}. \end{aligned} \tag{3.38}$$

Consequently, the only quantity that needs to be estimated is the baseline survival function $S_0(t) = \exp(-H_0(t))$. Rather than using an exact maximum likelihood estimator of $S_0(t)$, it is common to use the approximate estimator proposed by Breslow [40], which is defined as

$$\hat{S}_0(t) = \exp(-\hat{H}_0(t)) \tag{3.39}$$

$$\hat{H}_0(t) = \sum_{\{i \mid t_i \leq t\}} \frac{d_i}{\sum_{j \in \mathcal{R}_i} \exp(\boldsymbol{x}_j^\top \boldsymbol{\beta})}. \tag{3.40}$$

Note that Breslow's estimator of the cumulative baseline hazard function simplifies to the Nelson-Aalen estimator in eq. (2.20) when $\boldsymbol{\beta} = \mathbf{0}_p$.

### 3.2.4 Stratified Proportional Hazards Model

One of the limitations of Cox's proportional hazards model in eq. (3.20) is the proportional hazards assumption, i.e., for any two feature vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, the ratio of their corresponding hazard functions is constant for all time points. This assumption can be relaxed by dividing patients into $K$ non-overlapping groups, called *strata*, according to some observed characteristic such as age or sex. The coefficients still remain the same across all strata, but each stratum is characterized by an independent baseline hazard function $h_{0_k}(t)$. Formally, the hazard function of the $i$-th individual in stratum $k$ is defined as [176, pp. 308-312]:

$$h(t|\boldsymbol{x}_i, k) = h_{0_k}(t) \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right), \qquad k = 1, \ldots, K. \tag{3.41}$$

As a result, the proportional hazards assumption is only required to hold for subjects within the same stratum, but not for subjects in different strata. The simplest form of stratification splits patients according to one categorical variable or one continuous variable, after it has been discretized (e.g. age $< 60$ and age $\geq 60$). It is also possible to stratify on multiple variables, which results in one stratum for each combination of the categories from all variables. For instance, if one would stratify by sex and age, four strata would be created. The disadvantage of stratification is that the model cannot estimate a coefficient for the stratification variable anymore, which means its importance cannot be determined by the model.

Estimation of the coefficients in the stratified Cox's proportional hazards model can be carried out as described in section 3.2.2, with the only difference that risk sets $\mathcal{R}_i$ are now stratum specific. The partial likelihood function for $m$ strata is given by

$$PL_{\text{Stratified}}(\boldsymbol{\beta}) = \prod_{k=1}^{K} PL_k(\boldsymbol{\beta}), \tag{3.42}$$

where $PL_k(\boldsymbol{\beta})$ is the partial likelihood function in eq. (3.35) or (3.36) that only considers subjects in the $k$-th stratum [176, pp. 308-312].

### 3.2.5 Penalized Models

The classical Cox's proportional hazards model described in eq. (3.20) works well if there are no multicolinearities and the training data contains more samples than features. However, in many situations these conditions are violated. For instance, when building a survival model from gene expression data, the number of features is in the range of thousands, while the number of patients is usually below one thousand, and expression levels of some genes are highly correlated with each other. To overcome this limitation, the partial likelihood can be augmented by a $\ell_1$ (LASSO) or $\ell_2$ (ridge) penalty [288, 306]:

$$\log PL_{\text{LASSO}}(\boldsymbol{\beta}) = \log PL(\boldsymbol{\beta}) - \lambda_1 \sum_{j=1}^{p} |\beta_j| \tag{3.43}$$

$$\log PL_{\text{Ridge}}(\boldsymbol{\beta}) = \log PL(\boldsymbol{\beta}) - \frac{\lambda_2}{2} \sum_{j=1}^{p} \beta_j^2, \tag{3.44}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are hyper-parameters that determine the amount of regularization. Due to the non-differentiability at zero, the LASSO penalty effectively limits the complexity of the model by shrinking coefficients towards zero and setting some features exactly to zero, which makes it suitable for high-dimensional data and when $p > n$. Therefore, Cox's proportional hazards model with the LASSO penalty has feature selection embedded into its optimization. The hyper-parameter $\lambda_1$ controls the number of selected features – smaller values result in less non-zero coefficients. The ridge penalty is fully differentiable, therefore it only shrinks coefficients towards zero, but does not set them exactly to zero. It is most effective if survival data contains features that are highly correlated with each other [332].

One issue with a LASSO-penalized Cox model is that the regularization parameter $\lambda_1$ is applied to each coefficient equally, which leads to biased estimates of large coefficients if $\lambda_1$ is too big [328]. In addition, choosing a smaller value for $\lambda_1$ to avoid the bias may result in a model that is too complex, i.e., $\boldsymbol{\beta}$ is dense. The adaptive LASSO penalizes large coefficients less than small coefficients to reduce the bias of the LASSO [328]. In addition, it can be used to include prior knowledge about the importance of variables by penalizing important variables less than putative unimportant variables. The log partial likelihood of Cox's proportional hazards model with adaptive LASSO penalty is defined as

$$\log PL_{\text{A-LASSO}}(\boldsymbol{\beta}) = \log PL(\boldsymbol{\beta}) - \lambda_1 \sum_{j=1}^{p} w_j |\beta_j|, \tag{3.45}$$

where each feature is associated with a non-negative weight $w_j$.

Another disadvantage of the $\ell_1$ penalty is that it is only able to select $n$ features if the number of features $p$ exceeds the number of samples $n$ ($p > n$) [332]. Moreover, if data contains a group of features that are highly correlated, the $\ell_1$ penalty is going to randomly choose one feature from this group. To alleviate these problems, Zou and Hastie [332] proposed the elastic net penalty, which is a weighted combination between the $\ell_1$ and $\ell_2$ penalty:

$$\log PL_{\text{Elastic-net}}(\boldsymbol{\beta}) = \log PL(\boldsymbol{\beta}) - \lambda \left( \alpha \sum_{j=1}^{p} |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2 \right), \tag{3.46}$$

where $\lambda > 0$ and $\alpha \in [0; 1]$ is the ratio between the LASSO and ridge penalty.

The group LASSO penalty [327] behaves similar to the regular LASSO penalty, but couples the coefficients of a group of variables: either all variables of a group enter the model (non-zero coefficients), or all variables are excluded (zero coefficients). This behavior can be beneficial when a categorical variable is dummy coded and one wants to include or exclude all of the dummy variables. The group LASSO splits all $p$ features into $G$ non-overlapping groups and applies the $\ell_1$-norm to the $\ell_2$-norm of the coefficients of each individual group:

$$\log PL_{\text{G-LASSO}}(\boldsymbol{\beta}) = \log PL(\boldsymbol{\beta}) - \lambda_1 \sum_{g=1}^{G} \sqrt{|\mathcal{G}_g|} \sum_{j \in \mathcal{G}_g} \beta_j^2, \tag{3.47}$$

where the set $\mathcal{G}_g$ contains the indices of coefficients belonging to the $g$-th group.

### Optimization

The $\ell_2$-penalized log partial likelihood function (3.44) is convex and differentiable, hence the Newton-Rhapson method can be used to estimate the coefficients $\beta$ for fixed $\lambda_2$ (see section 3.2.2). In contrast, penalties with a $\ell_1$-norm are convex but non-differentiable, which means Newton-Rhapson optimization cannot be applied. Next, I will describe methods to optimize the log partial likelihood function with elastic net penalty (3.46), which contains the LASSO as a special case ($\alpha = 1$).

Moreover, a suitable value for $\lambda$ is often unknown in advance and has to be determined via cross-validation. Therefore, it would be computationally advantageous to efficiently compute coefficients along a path of $\lambda$ values. This can be achieved by starting with a high $\lambda$ value that sets all coefficients to zero and then incrementally decreasing its value until all coefficients enter the model unpenalized ($\lambda = 0$). By using the estimated coefficients $\hat{\boldsymbol{\beta}}$ of the previous value for $\lambda$ as initialization for the optimization with respect to the current $\lambda$, usually only a few iterations are required to find an updated estimate. This procedure is called *warm start* and can be utilized in any of the optimization schemes described below.

Park and Hastie [227] computed an entire solution path by exploiting the near piecewise linearity of the coefficients along the path of $\lambda$ values. They first approximated the change in the coefficients when decreasing $\lambda$, which was subsequently used as starting point for Newton's method; their technique is part of the predictor-corrector method class of algorithms for optimization. Goemann [113] solved eq. (3.46) by combining gradient ascent optimization with the Newton–Raphson algorithm. Simon *et al.* [267] used coordinate descent by reformulating eq. (3.46) as a weighted least squares problem. In coordinate descent, optimization is performed with respect to a single coefficient $\beta_j$, while keeping the remaining coefficients fixed, which allows solving for $\beta_j$ via the soft-thresholding operator, even if an $\ell_1$ penalty is used. This procedure is repeated multiple times for all coefficients until convergence. Alternatively, it is possible to apply general optimization algorithms that are suitable in the presence of convex and non-differentiable penalty functions, such as alternating direction method of multipliers (ADMM; [32, 106, 112]) or fast iterative shrinkage-thresholding algorithm (FISTA; [19]).

## 3.3 Support Vector Machines for Survival Analysis

The main idea of the support vector machine (SVM) for binary classification is that the hyperplane separating samples of the two classes should be positioned such that the closest distance to the hyperplane from either class is maximized [64]. By following similar ideas as in binary classification, Vapnik [305] and Herbrich *et al.* [138] applied the maximum margin principal to regression and learning-to-rank problems, respectively. In this section, I will describe three different extensions of support vector machines to model right censored survival data. Evers and Messow [90] and Van Belle *et al.* [294] cast it as a learning-to-rank problem by extending Rank SVM [138], Khan and Zubek [172] and Shivaswamy *et al.* [264] cast it as a regression problem, and Eleuteri [85] and Eleuteri and Taktak [86] as a quantile regression problem.

### 3.3.1 Using Ranking Constraints

The objective of Rank SVM is to learn a model that correctly ranks samples, grouped by query, according to their relevance value [138]. Each of the $n$ training samples consists of a triplet $(\boldsymbol{x}_i, q_i, r_i)$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ is a feature vector, $q_i \in Q \subset \mathbb{Z}$, a query, $r_i \in K \subset \mathbb{R}$ a relevance level, and $i = 1, \ldots, n$. During training, the coefficients $\boldsymbol{w} \in \mathbb{R}^p$ are learned such that the predicted ordering of samples, within each query, approximates the ordering according to the actual relevance levels (samples belonging to different queries are not compared).

**Definition 3.4: Linear Rank SVM.** Let $\mathcal{P} = \{(i, j) \mid q_i = q_j \wedge r_i > r_j\}_{i,j=1}^n$ be the set of comparable pairs, then linear Rank SVM optimizes the objective

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma \sum_{(i,j)\in\mathcal{P}} \xi_{ij} \\ \text{subject to} \quad & \boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \boldsymbol{x}_j \geq 1 - \xi_{ij}, \quad \forall (i,j) \in \mathcal{P}, \\ & \xi_{ij} \geq 0, \qquad\qquad\qquad\qquad \forall (i,j) \in \mathcal{P}, \end{aligned} \tag{3.48}$$

where $\gamma > 0$ is a regularization parameter and $\xi_{ij}$ are the slack variables. Alternatively, eq. (3.48) can be expressed as an unconstrained optimization problem as

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma \sum_{(i,j)\in\mathcal{P}} \max(0, 1 - \boldsymbol{w}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)). \tag{3.49}$$

With respect to survival analysis, training data consists of triplets $(\boldsymbol{x}_i, y_i, \delta_i)$ and the objective is to rank patients according to their survival time. Because of right censoring, the only observable quantity is $y_i = \min(t_i, c_i)$, where $t_i$ is the time of an event and $c_i$ the time of censoring. Therefore, a pairwise comparison is only valid if the patient with the lower observed time $y_i$ experienced an event (is uncensored). Formally, the set of comparable pairs $\mathcal{P}$ is given by $\mathcal{P} = \{(i,j) \mid y_i > y_j \land \delta_j = 1\}_{i,j=1}^n$, where it is assumed that all observed time points are unique [90, 294]. It is easy to see that the survival support vector machine simplifies to Rank SVM with a single query if all records are uncensored.

Optimization can be carried out by transforming the ranking objective into a classification objective and using a standard dual SVM solver [234]. This requires to explicitly construct all pairwise differences $\tilde{\boldsymbol{x}}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$ for all pairs $(i,j) \in \mathcal{P}$ together with the corresponding label

$$\tilde{y}_{ij} = \begin{cases} -1 & \text{if } y_i < y_j, \\ +1 & \text{if } y_i > y_j. \end{cases}$$

The disadvantage of this approach is that it requires $O(n^2)$ space and $O(pn^4)$ time for training, which is only feasible for small datasets. Van Belle *et al.* [295] addressed this problem by first clustering data according to observed times and limiting the number of comparable pairs. Instead of considering all uncensored samples $j$ as candidate in the pair $(i,j)$, they confined $j$ to uncensored samples that are among the $k$ nearest neighbors of $i$:

$$\mathcal{P}_{k\text{-NN}} = \{(i,j) \mid y_i > y_j \land \delta_j = 1 \land j \text{ is } k \text{ nearest neighbor of } i\}_{i,j=1}^n.$$

Setting $k = 1$ reduces the maximum number of constraints to $n$ and only maximizes the margin between adjacent comparable pairs. In this case, the $i$-th sample is only compared to the largest uncensored sample $j$ with $y_i > y_j$, which yields the set

$$\mathcal{P}_{1\text{-NN}} = \{(i,j) \mid y_i > y_j \land \delta_j = 1 \land \nexists k : y_i > y_k > y_j \land \delta_k = 1\}_{i,j=1}^n. \tag{3.50}$$

The work in [296] is based on the set $\mathcal{P}_{1\text{-NN}}$ and a modification of the maximum-margin constraint in eq. (3.48). Instead of maximizing the margin between comparable pairs, authors of [296] searched for a monotonically increasing transformation function $h$ with minimum Lipschitz constant $0 \leq l < \infty$ such that $h(\boldsymbol{w}^\top \boldsymbol{x}_i) - h(\boldsymbol{w}^\top \boldsymbol{x}_j) \leq l(\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \boldsymbol{x}_j)$. They showed that this corresponds to replacing 1 on the right side of the first constraint in eq. (3.48) by the term $y_i - y_j$, resulting in the optimization problem

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma \sum_{i=1}^n \xi_i \\ \text{subject to} \quad & \boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \boldsymbol{x}_j \geq y_i - y_j - \xi_i, && \forall (i,j) \in \mathcal{P}_{1\text{-NN}}, \\ & \xi_i \geq 0, && \forall i = 1, \ldots, n. \end{aligned} \tag{3.51}$$

By using $\mathcal{P}_{1\text{-NN}}$ instead of the full set in the minimization problem (3.51), the time complexity of training reduces from $O(pn^4)$ to $O(pn^2)$.

### 3.3.2 Using Regression Constraints

Instead of formulating survival analysis as a ranking problem, as described above, it can be addressed as a regression problem as well. In this case, models adapt the $\varepsilon$-insensitive loss used in support vector regression [305]. The idea is that the contribution of samples with small residuals, i.e., low prediction error, are ignored by the loss function.

**Definition 3.5: Linear Support Vector Regression.** The objective function for linear support vector regression is given by

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma \sum_{i=1}^{n} \Delta_\varepsilon(y_i - (\boldsymbol{w}^\top \boldsymbol{x}_i + b)), \tag{3.52}$$

where $\Delta_\varepsilon(r) = \max(0, |r| - \varepsilon)$, $\varepsilon > 0$ is the size of the insensitive region, $\gamma > 0$ is a regularization parameter, and $b \in \mathbb{R}$ is the intercept or bias term. Alternatively, the constrained optimization problem is defined as

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma \sum_{i=1}^{n} (\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - (\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq \varepsilon + \xi_i, \quad && \forall i = 1, \ldots, n, \\ & \boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i \leq \varepsilon + \xi_i^*, && \forall i = 1, \ldots, n, \\ & \xi_i, \xi_i^* \geq 0, && \forall i = 1, \ldots, n. \end{aligned} \tag{3.53}$$

where $\xi_i$ and $\xi_i^*$ are slack variables. The first constraint ensures that predictions that are at most $\varepsilon$ smaller than the actual value $y_i$ have no influence, and the second constraint that predictions that are at most $\varepsilon$ greater than $y_i$ have no influence.

For survival data, the exact time of an event is only known for individuals who have experienced an event ($\delta_i = 1$), but not for those censored ($\delta_i = 0$). Therefore, the constraints in eq. (3.53) have to be updated to consider right censored records. Shivaswamy *et al.* [264] proposed a modification of support vector regression for right censored survival data by disregarding the first constraint in eq. (3.53) for right censored observations, because the time of censoring only provides a lower bound on the time of an event. Similarly, for left censored records, the time of censoring provides an upper bound and therefore the second constraint in eq. (3.53) should be disregarded.

**Definition 3.6: SVCR.** Let $\mathcal{T} = \{i \mid \delta_i = 1\}$ denote the set of uncensored records, $\mathcal{C}_{\text{right}} = \{i \mid \delta_i = 0\}$ the set of right censored records and $\mathcal{C}_{\text{left}} = \{i \mid \delta_i = -1\}$ the set of left censored records. The training data can be split into two sets; first, $\mathcal{L} = \mathcal{T} \cup \mathcal{C}_{\text{right}}$ contains all subjects with a finite lower bound on survival time, and second, $\mathcal{U} = \mathcal{T} \cup \mathcal{C}_{\text{left}}$ contains all subjects with a finite upper bound. The objective of support vector regression for censored targets (SVCR; [264]) is defined as

$$\begin{aligned}
\min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma \left( \sum_{i \in \mathcal{L}} \xi_i + \sum_{j \in \mathcal{U}} \xi_j^* \right) \\
\text{subject to} \quad & y_i - (\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq \varepsilon + \xi_i, && \forall i \in \mathcal{L}, \\
& \boldsymbol{w}^\top \boldsymbol{x}_j + b - y_j \leq \varepsilon + \xi_j^*, && \forall j \in \mathcal{U}, \\
& \xi_i \geq 0, && \forall i \in \mathcal{L}, \\
& \xi_j^* \geq 0, && \forall j \in \mathcal{U}.
\end{aligned} \tag{3.54}$$

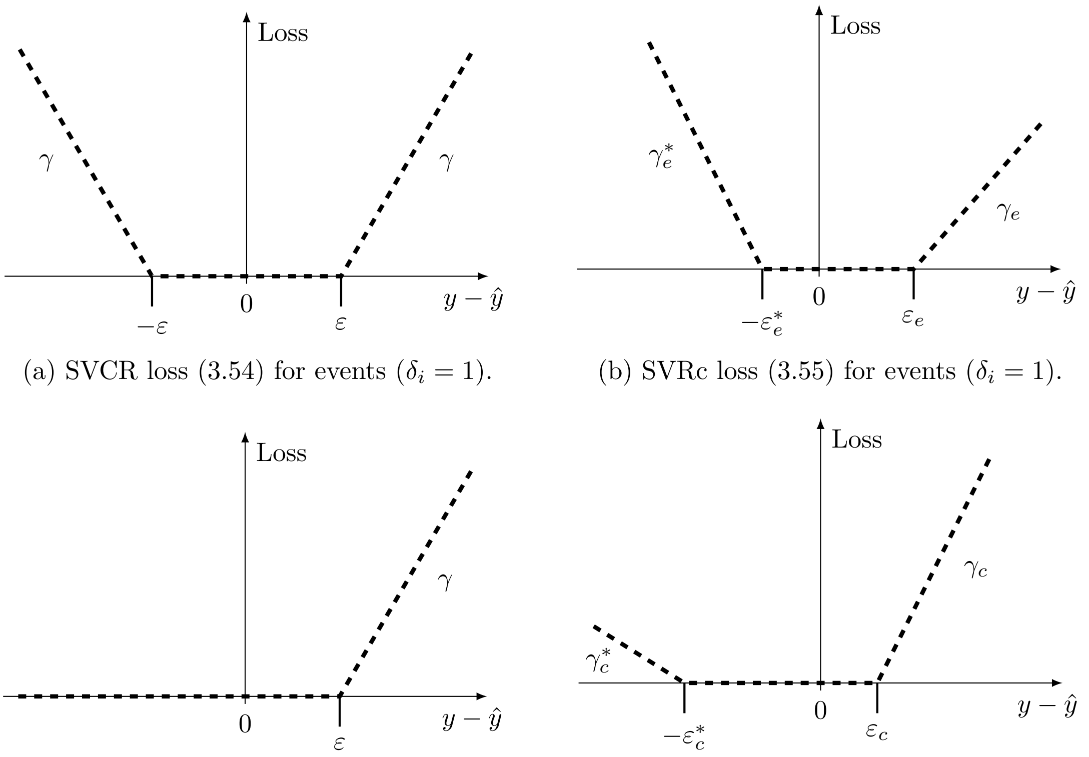
If there are no censored records, $\mathcal{T} = \mathcal{L} = \mathcal{U}$ and the objective reduces to the one of regular support vector regression in eq. (3.53).

An alternative formulation based on support vector regression was proposed by Khan and Zubek [172]. In contrast to the approach proposed in [264], they introduced an assymmetrical $\varepsilon$-insensitive loss with different margins for censored and uncensored patients. Instead of a single parameter $\varepsilon$, the objective function relies on four different parameters: $\varepsilon_e, \varepsilon_e^* > 0$ define the upper and lower bound of the insensitive area for uncensored records, and $\varepsilon_c, \varepsilon_c^* > 0$ define the upper and lower bound of the insensitive area for censored records. In addition, the regularization parameter $\gamma$ is replaced by four parameters, one for each type of error: 1) $\gamma_e > 0$ for uncensored records with predicted time $\hat{y}_i$ less than the actual survival time $y_i$, 2) $\gamma_e^* > 0$ for uncensored records with $\hat{y}_i > y_i$, 3) $\gamma_c > 0$ for censored records with $\hat{y}_i < y_i$, and 4) $\gamma_c^* > 0$ for censored records with $\hat{y}_i > y_i$.

**Definition 3.7: SVRc.** Let $s_i = I(\delta_i \neq 0)$ denote whether a patient's record is censored, then $\gamma_i = s_i\gamma_c + (1 - s_i)\gamma_e$ denotes the conditional regularization parameter for predictions that are less than the actual survival time with respect to the upper bound $\varepsilon_i = s_i\varepsilon_c + (1 - s_i)\varepsilon_e$. Analogously, $\gamma_i^*$ denotes the conditional regularization parameter for predictions that are greater than the actual survival time with respect to the lower bound $\varepsilon_i^*$. The objective of support vector regression for censored data (SVRc; [172]) is defined as

$$\begin{aligned}
\min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i=1}^{n} (\gamma_i \xi_i + \gamma_i^* \xi_i^*) \\
\text{subject to} \quad & y_i - (\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq \varepsilon_i + \xi_i, && \forall i = 1, \ldots, n, \\
& \boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i \leq \varepsilon_i^* + \xi_i^*, && \forall i = 1, \ldots, n, \\
& \xi_i, \xi_i^* \geq 0, && \forall i = 1, \ldots, n.
\end{aligned} \tag{3.55}$$

Note that SVCR in eq. (3.54) does not penalize predictions that are greater than the observed time of censoring for right censored samples, whereas SVRc in eq. (3.55) penalizes predictions that exceed the actual time of censoring by more than $\varepsilon_c^*$ by $\gamma_c^*$. Therefore, Khan and Zubek [172] suggested that predictions greater than the time of



(a) SVCR loss (3.54) for events ($\delta_i = 1$).

(b) SVRc loss (3.55) for events ($\delta_i = 1$).

(c) SVCR loss (3.54) for censored samples ($\delta_i = 0$).

(d) SVRc loss (3.55) for censored samples ($\delta_i = 0$).

**Figure 3.1**: Graphical representation of loss functions used by SVCR [264] in eq. (3.54) and SVRc [172] in eq. (3.55). For uncensored records, SVCR uses a symmetric $\varepsilon$-insensitive loss function (a), whereas SVRc penalizes predictions that exceed the actual survival time by more than $\varepsilon_e^*$ or that are at least $\varepsilon_e$ below the actual survival time (b). For censored records, SVCR does not penalize predictions that are greater than the time of censoring (c). In contrast, SVRc penalizes predictions that exceed the time of censoring by more than $\varepsilon_c^*$ (d). For SVRc, Khan and Zubek [172] suggest that $\varepsilon_c^* > \varepsilon_e > \varepsilon_e^* = \varepsilon_c$ and $\gamma_c^* < \gamma_e < \gamma_e^* = \gamma_c$, which holds in the example shown here too.

right censoring should be penalized less than predictions smaller than the time of right censoring. They recommended the following relationship between the hyper-parameters: $\varepsilon_c^* > \varepsilon_e > \varepsilon_e^* = \varepsilon_c$ and $\gamma_c^* < \gamma_e < \gamma_e^* = \gamma_c$. Figure 3.1 illustrates the differences between the loss function employed by Shivaswamy *et al.* [264] and Khan and Zubek [172].

Both the optimization problem (3.54) and (3.55) have $O(n)$ constraints, which is less than in the case of survival support vector machine with ranking constraints in section 3.3.1.

### 3.3.3 Using Hybrid Ranking and Regression Constraints

One disadvantage of models based on ranking constraints described in section 3.3.1 is that they only provide a relative ordering of patients with respect to their survival time in the prediction phase. For a given set of patients, the model only predicts in which order subjects are expected to experience an event, but it does not provide any information about the time of an event. The latter is only captured by regression models presented in section 3.3.2. Van Belle *et al.* [297] combined models (3.51) and (3.54) to obtain a hybrid model, which includes both ranking and regression constraints:

$$\begin{aligned}
\min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma \left( \sum_{i \in \mathcal{L}} \xi_i + \sum_{j \in \mathcal{U}} \xi_j^* \right) + \theta \sum_{i=1}^{n} \zeta_i \\
\text{subject to} \quad & y_i - (\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq \varepsilon + \xi_i, && \forall i \in \mathcal{L}, \\
& \boldsymbol{w}^\top \boldsymbol{x}_j + b - y_j \leq \varepsilon + \xi_j^*, && \forall j \in \mathcal{U}, \\
& \boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \boldsymbol{x}_j \geq y_i - y_j - \zeta_i, && \forall i \in \mathcal{P}_{\text{1-NN}}, \\
& \xi_i \geq 0, && \forall i \in \mathcal{L}, \\
& \xi_j^* \geq 0, && \forall j \in \mathcal{U}, \\
& \zeta_i \geq 0, && \forall i = 1, \ldots, n.
\end{aligned} \tag{3.56}$$

The optimization problem (3.56) has $O(n)$ constraints due to using $\mathcal{P}_{\text{1-NN}}$ in the ranking constraint.

### 3.3.4 Using Quantile Regression Loss

Let $T$ denote a continuous non-negative random variable corresponding to a patient's survival time with cumulative distribution function $F_T(t) = P(T \leq t)$, and let $\kappa \in [0; 1]$. The $\kappa$-th quantile of $T$, denoted by $Q_T(\kappa)$, is given by

$$Q_T(\kappa) = F_T^{-1}(\kappa) = \inf\{t \in T \mid F_T(t) \geq \kappa\}.$$

From the definition of the survival function (2.2), it follows that the $\kappa$-th quantile of $T$ is identical to the $(1-\kappa)$-th quantile of $S(t) = 1 - F(t)$, which suggests estimating the survival function $S(t)$ via quantile regression.

> **Definition 3.8: Quantile Regression.** In quantile regression [177, 178], the objective is to estimate the conditional quantile function $Q_{T|X}(\kappa, \boldsymbol{x})$ for a pair of continuous random variables $(T, X)$ with unknown joint distribution, where $\boldsymbol{x} \in X = X_1 \times \cdots \times X_p$. The conditional quantile function $Q_{T|X}(\kappa, \boldsymbol{x}) : X \to \mathbb{R}$ is given by
>
> $$Q_{T|X}(\kappa, \boldsymbol{x}) = F_T^{-1}(\kappa | X = \boldsymbol{x}) = \inf\{t \in T \mid F_T(t|X = \boldsymbol{x}) \geq \kappa\},$$
>
> where $F_T(\cdot|X = \boldsymbol{x})$ is the conditional cumulative distribution function of $T$, given $X = \boldsymbol{x}$.

Koenker and Bassett [178] showed that the $\kappa$-th quantile of a sample $y_1, \ldots y_n$ can be estimated by

$$\min_{q \in \mathbb{R}} \quad \sum_{i=1}^{n} \rho_\kappa(y_i - q), \tag{3.57}$$

where

$$\rho_\kappa(r) = \begin{cases} (\kappa - 1)r & \text{if } r < 0, \\ \kappa r & \text{if } r \geq 0, \end{cases}$$

which has been adopted by Takeuchi *et al.* [282] to yield their empirical conditional quantile estimator:

$$\min_{\boldsymbol{w}, b} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{\gamma}{n} \sum_{i=1}^{n} \rho_\kappa(y_i - (\boldsymbol{w}^\top \boldsymbol{x}_i + b)). \tag{3.58}$$

Equation (3.58) can be expressed as a constrained optimization problem as

$$\begin{aligned} \min_{\boldsymbol{w}, b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{\gamma}{n} \sum_{i=1}^{n} \kappa \xi_i + (1-\kappa)\xi_i^* \\ \text{subject to} \quad & y_i - (\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq \xi_i, && \forall i = 1, \ldots, n, \\ & \boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i \leq \xi_i^*, && \forall i = 1, \ldots, n, \\ & \xi_i, \xi_i^* \geq 0, && \forall i = 1, \ldots, n. \end{aligned} \tag{3.59}$$

In [85, 86], the authors extended the constrained optimization problem (3.59) to be applicable to right censored survival data – under the assumption that survival time and censoring time are conditionally independent, given the features – by introducing an *inverse probability of censoring weighted* (IPCW) loss function [49, 240, 301].

**Definition 3.9: Quantile Regression for Right Censored Survival Data via Support Vector Machines.** Let $T$ and $C$ denote continuous random variables representing the survival time and censoring time, and let $\boldsymbol{x}$ be a $p$-dimensional feature vector. In practice, it is only possible to observe $Y = \min(T, C)$ and the event indicator $\delta = I(T \leq C)$. The estimator (3.59) can be extended to partially observed survival data by assigning each sample a weight $\omega_i$ proportional to the inverse probability of being censored after time $y_i$, given the feature vector $\boldsymbol{x}$ (see section 3.1.2 on page 34). By including weights $\omega_i$ into the objective (3.59) and applying a logarithmic transformation to the observed time points $Y$, the objective becomes

$$\begin{aligned} \min_{\boldsymbol{w}, b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{\gamma}{n}\sum_{i=1}^{n} \omega_i \left[\kappa \xi_i + (1-\kappa)\xi_i^*\right] & & \\ \text{subject to} \quad & \log y_i - (\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq \xi_i, & & \forall i = 1, \ldots, n, \\ & \boldsymbol{w}^\top \boldsymbol{x}_i + b - \log y_i \leq \xi_i^*, & & \forall i = 1, \ldots, n, \\ & \xi_i, \xi_i^* \geq 0, & & \forall i = 1, \ldots, n. \end{aligned} \tag{3.60}$$

Eleuteri [85] assumed that censoring is independent of the features and used the non-parametric Kaplan-Meier estimator to compute the weights $\omega_i$.

*Note.* Quantile regression models for censored data have been proposed outside of support vector machines as well (e.g. [101]).

## 3.3.5 Non-linear Extension

The support vector machine based on ranking constraints (see section 3.3.1), regression constraints (see section 3.3.2), and quantile regression loss (see section 3.3.4) all assume a linear decision function $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ with coefficients $\boldsymbol{w} \in \mathbb{R}^p$ and constant bias term $b \in \mathbb{R}$. However, if data becomes increasingly more complex, a non-linear decision function is often preferred. To obtain a non-linear decision function in the support vector machine framework, feature vectors are mapped into a higher dimensional (possible infinite dimensional) Hilbert space $\mathcal{H}$ via the mapping function $\phi$, defined as $\phi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{X}$ is the domain of $\boldsymbol{x}$ (usually $\mathcal{X} \subset \mathbb{R}^p$). Performing training on the transformed data results in a linear decision function in the Hilbert space $\mathcal{H}$: $f_{\mathcal{H}}(\boldsymbol{x}) = \boldsymbol{w}_{\mathcal{H}}^\top \phi(\boldsymbol{x}) + b$.

**Example 3.2.** If the mapping function from $\mathbb{R}^2$ to $\mathbb{R}^3$ is defined as $\phi((x_1, x_2)^\top) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^\top$, the decision function becomes

$$f_{\mathcal{H}}(\boldsymbol{x}) = \boldsymbol{w}_{\mathcal{H}}^\top \phi(\boldsymbol{x}) + b = w_1 x_1^2 + w_2\sqrt{2}x_1x_2 + w_3 x_2^2 + b.$$

The decision boundary given by $f_{\mathcal{H}}(\boldsymbol{x}) = 0$ represents a conic section (ellipse, parabola or hyperbola) in $\mathbb{R}^2$. For instance, with $\boldsymbol{w}_{\mathcal{H}} = (1, 0, 1)^\top$ and $b = -1$, the decision boundary is a unit circle:

$$f_{\mathcal{H}}(\boldsymbol{x}) = 0 \Leftrightarrow x_1^2 + x_2^2 = 1.$$

Obviously, explicitly transforming data into a higher-dimensional space, especially if it is infinite dimensional, can be prohibitive. This can be avoided by using the Kernel trick [5, 31], which exploits that a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ directly computes the inner product in $\mathcal{H}$:

$$k(\boldsymbol{x}, \boldsymbol{z}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle_{\mathcal{H}}$$

The kernel function $k$ gives rise to a reproducing kernel Hilbert space (RKHS) of functions $f : \mathcal{X} \to \mathbb{R}$, denoted as $\mathcal{H}_k$. Thus, the minimization problems from above can be extended to the non-linear case by solving

$$\min_{f \in \mathcal{H}_k} \quad \frac{1}{2} \|f\|^2_{\mathcal{H}_k} + \gamma \sum_{i=1}^{n} L(y_i, \delta_i, f(\boldsymbol{x}_i) + b), \tag{3.61}$$

where $\|\cdot\|_{\mathcal{H}_k}$ is a norm in $\mathcal{H}_k$, induced by the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_k}$, and $L$ is one of the convex loss functions used in eqs. (3.48), (3.51), (3.54) to (3.56) and (3.59). It can be shown [173, 307] that under mild conditions there exists a finite-dimensional solution to (3.61) of the form

$$f(\boldsymbol{z}) = \sum_{i=1}^{n} \beta_i k(\boldsymbol{x}_i, \boldsymbol{z}), \tag{3.62}$$

which is known as the representer theorem [173].

### Dual Optimization

Traditionally, training maximum-margin models consists of solving a constrained optimization problem that involves a convex loss function, which is called the *primal* objective function. A primal optimization problem is converted into a *dual* optimization problem by augmenting the primal objective function with additional terms representing the constraints, the so-called *Lagrange multipliers* (see e.g. [33]). In the context of support vector machines, the dual problem can usually be expressed in form of inner products of the training data, which in turn enables the use of kernel functions. I will demonstrate this technique based on the survival support vector machine with ranking constraints in eq. (3.48).

The first step consists of constructing the Lagrangian primal function $L_p$ of (3.48):

$$\begin{aligned} L_P(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\alpha}') = & \frac{1}{2} \|\boldsymbol{w}\|_2^2 + \gamma \sum_{(i,j) \in \mathcal{P}} \xi_{ij} \\ & - \sum_{(i,j) \in \mathcal{P}} \alpha_{ij} (\boldsymbol{w}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) - 1 + \xi_{ij}) - \sum_{(i,j) \in \mathcal{P}} \alpha'_{ij} \xi_{ij}, \end{aligned} \tag{3.63}$$

where $\alpha_{ij}, \alpha'_{ij} \geq 0$ are the dual variables. The Karush-Kuhn-Tucker conditions [33, 169, 179] state sufficient conditions for which the minimum of (3.48) with respect to $\boldsymbol{w}$

coincides with the maximum of (3.63) with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ (the duality gap is zero), they are:

$$
\begin{aligned}
\frac{\partial L_P(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\alpha}')}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{(i,j) \in \mathcal{P}} \alpha_{ij} (\boldsymbol{x}_i - \boldsymbol{x}_j) = 0, && \quad (3.64)\\
\frac{\partial L_P(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\alpha}')}{\partial \xi_{ij}} = \gamma - \alpha_{ij} - \alpha'_{ij} = 0, && \quad (3.65)\\
\boldsymbol{w}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) - 1 + \xi_{ij} \geq 0, \qquad \forall (i,j) \in \mathcal{P}, && \quad (3.66)\\
\xi_{ij} \geq 0, \qquad \forall (i,j) \in \mathcal{P}, && \quad (3.67)\\
\alpha_{ij} (\boldsymbol{w}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) - 1 + \xi_{ij}) = 0, \qquad \forall (i,j) \in \mathcal{P}, && \quad (3.68)\\
\alpha'_{ij} \xi_{ij} = 0, \qquad \forall (i,j) \in \mathcal{P}, && \quad (3.69)\\
\alpha_{ij} \geq 0, \qquad \forall (i,j) \in \mathcal{P}, && \quad (3.70)\\
\alpha'_{ij} \geq 0, \qquad \forall (i,j) \in \mathcal{P}. && \quad (3.71)
\end{aligned}
$$

Plugging eqs. (3.64) and (3.69) into (3.63), parameters $\boldsymbol{w}$, $\boldsymbol{\xi}$ and $\boldsymbol{\alpha}'$ can be eliminated and the result is the Lagrangian dual function

$$
\begin{aligned}
L_D(\boldsymbol{\alpha}) =& \frac{1}{2} \sum_{(i,j) \in \mathcal{P}} \sum_{(u,v) \in \mathcal{P}} \alpha_{ij} \alpha_{uv} \langle \boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_u - \boldsymbol{x}_v \rangle \\
& - \sum_{(i,j) \in \mathcal{P}} \sum_{(u,v) \in \mathcal{P}} \alpha_{ij} \alpha_{uv} \langle \boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_u - \boldsymbol{x}_v \rangle + \sum_{(i,j) \in \mathcal{P}} \alpha_{ij} \\
=& \sum_{(i,j) \in \mathcal{P}} \alpha_{ij} - \frac{1}{2} \sum_{(i,j) \in \mathcal{P}} \sum_{(u,v) \in \mathcal{P}} \alpha_{ij} \alpha_{uv} \langle \boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_u - \boldsymbol{x}_v \rangle .
\end{aligned} \qquad (3.72)
$$

Equation (3.72) can be written in matrix form by introducing a $|\mathcal{P}| \times n$ sparse matrix $\boldsymbol{A}$ that encodes comparable pairs of samples. For each pair in $\mathcal{P}$, there is one row in $\boldsymbol{A}$ that is all zero except for two entries corresponding to the associated pair in $\mathcal{P}$:

$$
(i,j) \in \mathcal{P} \Rightarrow \exists k \in \{1, \ldots, m\} \mid A_{kl} = \begin{cases} 1 & \text{if } l = i, \\ -1 & \text{if } l = j, \\ 0 & \text{else,} \end{cases} \qquad (3.73)
$$

where $l = 1, \ldots, n$ and $m = |\mathcal{P}|$. Based on the above definition, the dual optimization problem for linear survival support vector machines with ranking constraints (3.72) can be rewritten as

$$
\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \mathbb{1}_m - \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{A} \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{A}^\top \boldsymbol{\alpha} \\
\text{subject to} \quad & 0 \leq \alpha_{ij} \leq \gamma, \quad \forall (i,j) \in \mathcal{P},
\end{aligned} \qquad (3.74)
$$

where $\boldsymbol{\alpha} \in \mathbb{R}^m$ are the coefficients and the constraints are due to eqs. (3.65) and (3.71).

The extension to the non-linear case is straightforward by replacing every $\boldsymbol{x}$ with its transformation $\phi(\boldsymbol{x})$ and noting that $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle_{\mathcal{H}_k} = k(\boldsymbol{x}, \boldsymbol{z})$. Let $\boldsymbol{K}$ denote the $n \times n$ kernel matrix with entries $\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, then the dual optimization problem for non-linear survival support vector machines with ranking constraints is

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \mathbb{1}_m - \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{A} \boldsymbol{K} \boldsymbol{A}^\top \boldsymbol{\alpha} \\ \text{subject to} \quad & 0 \leq \alpha_{ij} \leq \gamma, \quad \forall (i,j) \in \mathcal{P}. \end{aligned} \tag{3.75}$$

Given a new set of data points $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N$, the $k$-th predicted risk score can be computed by

$$\hat{f}(\boldsymbol{z}_k) = \sum_{(i,j) \in \mathcal{P}_{\mathrm{SV}}} \alpha_{ij} (k(\boldsymbol{x}_i, \boldsymbol{z}_k) - k(\boldsymbol{x}_j, \boldsymbol{z}_k)),$$

where $\mathcal{P}_{\mathrm{SV}}$ is the set of pairs for which $\alpha_{ij} > 0$, i.e., the support pairs. Let $\boldsymbol{K}^* \in \mathbb{R}^{N \times n}$ be the kernel matrix with entries $\boldsymbol{K}^*_{i,j} = k(\boldsymbol{z}_i, \boldsymbol{x}_j)$, then the prediction can be expressed in matrix form as

$$\hat{f}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N) = \boldsymbol{K}^* \boldsymbol{A}_{\mathrm{SV}}^\top \boldsymbol{\alpha}_{\mathrm{SV}}, \tag{3.76}$$

where $\boldsymbol{\alpha}_{\mathrm{SV}}$ is a vector of $m_{\mathrm{SV}}$ non-zero dual coefficients and $\boldsymbol{A}_{\mathrm{SV}} \in \{-1, 0, 1\}^{m_{\mathrm{SV}}, n}$ is the matrix $\boldsymbol{A}$ used during training, but restricted to rows corresponding to pairs in $\mathcal{P}_{\mathrm{SV}}$.

*Note.* Instead of storing $m_{\mathrm{SV}}$ dual coefficients, it is possible to just store $n$ coefficients of $\boldsymbol{A}_{\mathrm{SV}}^\top \boldsymbol{\alpha}_{\mathrm{SV}}$. Consequently, the prediction step requires $nN$ evaluations of the kernel function and $nN$ operations to compute all risk scores.

### Primal Optimization

Although a non-linear decision function is commonly obtained by performing optimization in the dual, Chapelle [53] showed that a non-linear extension is also possible by considering an unconstrained optimization problem and applying the representer theorem directly. For a convex loss function $L$, one has to solve

$$\min_{\boldsymbol{\beta}, b} \quad \boldsymbol{\beta}^\top \boldsymbol{K} \boldsymbol{\beta} + \gamma \sum_{i=1}^{n} L(y_i, \boldsymbol{K}_i \boldsymbol{\beta} + b), \tag{3.77}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)^\top$ is the vector of coefficients, and $\boldsymbol{K}_i$ denotes the $i$-th row vector of the kernel matrix $\boldsymbol{K}$.

Regarding the unconstrained optimization problem (3.49) for survival support vector machines with ranking constraints, the objective with a non-linear decision boundary becomes

$$\min_{\boldsymbol{\beta}} \quad \boldsymbol{\beta}^\top \boldsymbol{K} \boldsymbol{\beta} + \gamma \sum_{(i,j) \in \mathcal{P}} \max(0, 1 - \boldsymbol{K}_i \boldsymbol{\beta} + \boldsymbol{K}_j \boldsymbol{\beta}). \tag{3.78}$$

The prediction step simply becomes

$$\hat{f}(\boldsymbol{z}) = \sum_{i=1}^{n} \beta_i k(\boldsymbol{x}_i, \boldsymbol{z}),$$

which corresponds to the result (3.62) of the representer theorem.

*Note.* Converting a linear model to a non-linear model via the kernel trick demonstrated above is not limited to maximum-margin models. For instance, Cox's proportional hazards model has been extended to the non-linear case following the same ideas [48, 192].

### Mercer's Conditions

Now the question arises which kernel functions can be used to efficiently compute the inner product? The answer can be derived from Mercer's theorem and the theory of integral equations [212].

**Theorem 3.1: Mercer's conditions [66, 305].** *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be an inner product in some Hilbert space $\mathcal{H}$. The function $k$ can be represented as*

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{q=1}^{\infty} a_q \phi_q(\boldsymbol{x}_i) \phi_q(\boldsymbol{x}_j), \tag{3.79}$$

*with positive coefficients $a_q$ and linearly independent functions $\phi_q(\boldsymbol{x})$, if the necessary and sufficient condition*

$$\int_{\mathcal{X}} \int_{\mathcal{X}} k(u, v) g(u) g(v) du dv \geq 0 \tag{3.80}$$

*holds for all functions $g \neq 0$ with finite $L_2$ norm*

$$\|g\|_{L_2}^2 = \int_{\mathcal{X}} g^2(u) du < \infty.$$

In other words, if Mercer's conditions hold, $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ corresponds to a dot product in some higher dimensional (possible infinite dimensional) feature space $\mathcal{H}$. The positivity condition (3.80) is equivalent to the condition that the kernel matrix $\boldsymbol{K}$ with entries $\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive semidefinite for any collection of feature vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ [277].

Table 3.1 provides an overview of popular kernel functions. The hyperbolic tangent or sigmoid kernel deserves special consideration, because it is in fact a kernel that does not satisfy Mercer's conditions, but still has been used successfully in many applications. The problem was first observed by [305] and later on proven in [226, 270]. It also sparked research in the area of learning with indefinite kernels [201, 224].

**Table 3.1**: Examples of common kernel functions. †: Although, the sigmoid kernel was considered a valid kernel function in the beginning, it can be shown that it is not a kernel function that satisfies Mercer's conditions for all parameters $c$ and $\kappa$ [226, 270].

| Kernel | Definition | Parameters |
|---|---|---|
| Polynomial | $(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + c)^d$ | $d \in \mathbb{N}, c \geq 0$ |
| Hyperbolic tangent/Sigmoid† | $\tanh(c + \kappa \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle)$ | $c, \kappa \in \mathbb{R}$ |
| Radial basis function | $\exp(-\lVert \boldsymbol{x}_i - \boldsymbol{x}_j \rVert_2^2 / (2\sigma^2))$ | $\sigma > 0$ |

**Example 3.3.** Let us consider a polynomial kernel of degree two ($d = 2$, $c = 0$) applied to two feature vectors $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^2$ and the mapping function $\phi$ defined in example 3.2 above. By explicitly computing the result of the polynomial kernel, the connection to example 3.2, where a mapping function was defined explicitly, will become apparent:

$$\begin{aligned} k(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle^2 = (x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 \\ &= (x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2)(x_{j1}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2)^\top \\ &= \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle. \end{aligned}$$

The example shows that a polynomial kernel of degree two maps feature vectors into a finite three dimensional Euclidean space, where the inner product is computed. In the general case, any homogeneous polynomial kernel of degree $d$ projects data from a $p$-dimensional space into a Euclidean space $\mathcal{H}$ of dimension $\binom{p+d-1}{d}$ [46, Theorem 4].

Instead of proofing Mercer's conditions for every kernel, it is possible to combine existing kernel functions to form a new kernel function satisfying Mercer's conditions. Corollaries in [271] show that positive linear combinations of kernels, products of kernels and integrals of kernels are permitted.

**Corollary 3.1.** *Let $k_1$ and $k_2$ be kernel functions satisfying Mercer's conditions and $c_1, c_2 \geq 0$ two constants, then*

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = c_1 k_1(\boldsymbol{x}_i, \boldsymbol{x}_j) + c_2 k_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

*satisfies Mercer's conditions [271, Corollary 3].*

**Corollary 3.2.** *Let $k_1$ and $k_2$ be kernel functions satisfying Mercer's conditions, then*

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k_1(\boldsymbol{x}_i, \boldsymbol{x}_j) k_2(\boldsymbol{x}_i, \boldsymbol{x}_j),$$

*satisfies Mercer's conditions [271, Corollary 5].*

**Corollary 3.3.** *Let $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$ be a function on $\mathcal{X} \times \mathcal{X}$ such that*

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \int_{\mathcal{X}} s(\boldsymbol{x}_i, \boldsymbol{u}) s(\boldsymbol{x}_j, \boldsymbol{u}) d\boldsymbol{u},$$

*then $k$ satisfies Mercer's conditions [271, Corollary 4].*

### Clinical Kernel

General kernel functions as those presented in table 3.1 often yield good results when the scale of features is similar, otherwise features with generally higher values would dominate features with lower values in the computation of the kernel. Therefore, it is often suggested to standardize features to have mean 0 and variance 1. Clinical data is often a heterogeneous mix of features with different statistical properties. For instance, age is usually below 100, ethnicity is a categorical variable with multiple levels, and the concentration of total cholesterol in the blood is in the range of 130 to 200 mg/dl for healthy patients [278]. In particular, the numerical coding of categorical variables determines how similar two feature vectors are. Assuming ethnicity can take on any of the four values White, Black, Asian and Other, which are coded from zero to four, then a White person would be considered more similar to a Black person than to an Asian person. In general, no apparent ordering of races exists that justifies such a similarity. In contrast, similarities based on ordered categorical variables, such as low, medium, high, behave more like continuous variables. Consequently, employing standard kernel functions would lead to an inadequate measure of similarity.

Daemen *et al.* [70] addressed this problem by defining a kernel function that explicitly distinguishes between continuous, ordinal and categorical features. They proposed an additive kernel composed of $p$ kernel functions that yield similarities in the interval $[0, 1]$:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{p} \sum_{v=1}^{p} k_v(x_{iv}, x_{jv}), \tag{3.81}$$

where the definition of kernel function $k_v$ depends on the type of the $v$-th feature. For categorical features, the kernel function between the $i$-th and $j$-th subject's value is defined as the Kronecker delta function:

$$k_{\text{unorderd}}(x, z) = \begin{cases} 1 & \text{if } x = z, \\ 0 & \text{else,} \end{cases} \tag{3.82}$$

and for continuous and ordinal variables the kernel function becomes

$$k_{\text{ordered}}(x, z) = \frac{r - |x - z|}{r}, \tag{3.83}$$

where $r$ denotes the range of continuous values or the number of categories minus 1 for ordinal variables. The range $r$ can either be estimated from data or can be specified based on prior knowledge such as the maximum grade in the Gleason scoring system for prostate cancer [111].

## 3.4 Boosting

I will begin this section describing gradient boosting as a general concept without explicitly specifying a loss function to minimize. For simplicity, I will consider an arbitrary loss that only depends on one variable $y$ in sections 3.4.1 and 3.4.3. In section 3.4.4, I will explain how gradient boosting can be used for survival analysis by defining suitable loss functions for right censored survival data.

### 3.4.1 Functional Gradient Descent

Gradient boosting is a versatile framework for optimizing arbitrary loss functions via *functional gradient descent* [104]. With fully uncensored data consisting of feature vectors $\boldsymbol{x}$ and response $y$, the objective is to find a function $f$ that minimizes the expected loss with respect to a loss function $L(y, \hat{y})$:

$$\operatorname*{argmin}_{f(\cdot)} E\left[L(y, f(\boldsymbol{x}))\right], \tag{3.84}$$

where the expectation is over the joint distribution over all $(\boldsymbol{x}, y)$-pairs. In gradient boosting, the function $f$ is assumed to be an additive model of the form

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} \beta_m g(\boldsymbol{x}; \boldsymbol{\theta}_m),$$

where $M > 0$ denotes the size of the ensemble, and $\beta_m \in \mathbb{R}$ is a weighting term. The function $g$ is called a *base learner* and is parameterized by the vector $\boldsymbol{\theta}$. Individual base learners of $f$ differ in the configuration of their parameters $\boldsymbol{\theta}$, which is indicated by a subscript $m$.

To solve eq. (3.84), gradient boosting is extending the concept of steepest descent in parameter space to the function space. Algorithm 3.2 summarizes the traditional steepest descent algorithm to find coefficients $\boldsymbol{w}$ that minimize a differentiable loss function $\Phi(\boldsymbol{w})$. It is easy to see that the final solution $\boldsymbol{w}$ is constructed from summing up $M$ individual steps or "boosts."

Gradient boosting obtains the solution to eq. (3.84) by solving the following empirical risk minimization problem based on $n$ training samples:

$$\operatorname*{argmin}_{f(\cdot)} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\boldsymbol{x}_i)). \tag{3.85}$$

Instead of optimizing with respect to $\boldsymbol{w}$ in Euclidean space, gradient boosting considers the function $f(\boldsymbol{x})$ to be a parameter and applies gradient descent in function space. Algorithm 3.3 provides the details of the gradient boosting algorithm.

**Algorithm 3.2:** Steepest descent in Euclidean space.

```
Input: Differentiable loss function $\Phi$, number of steps $M > 0$, starting point $\boldsymbol{w}^{(0)}$.
Output: Coefficients $\boldsymbol{w} = \sum_{m=0}^{M} \rho_m \Delta \boldsymbol{w}^{(m)}$.
for $m \leftarrow 1$ to $M$ do
    Compute negative gradient and evaluate it at $\boldsymbol{w}^{(m-1)}$ to obtain search direction $\Delta \boldsymbol{w}^{(m)}$:
        $\Delta w_j^{(m)} = -\left[\frac{\partial \Phi(\boldsymbol{w})}{\partial w_j}\right]_{\boldsymbol{w}=\boldsymbol{w}^{(m-1)}}, \quad \forall j = 1, \ldots, p.$
    Determine step size $\rho_m > 0$ along direction $\Delta \boldsymbol{w}^{(m)}$ via line search:
        $\rho_m = \operatorname*{argmin}_{\rho} \Phi\left(\boldsymbol{w}^{(m-1)} + \rho \Delta \boldsymbol{w}^{(m)}\right).$
    Update coefficients: $\boldsymbol{w}^{(m)} = \boldsymbol{w}^{(m-1)} + \rho_m \Delta \boldsymbol{w}^{(m)}$.
end
```

It leverages the additive structure of the function $f(\boldsymbol{x})$ by constructing it in a greedy stagewise manner instead of optimizing it with respect to all base learners concurrently, i.e., it performs

$$(\hat{\beta}_m, \hat{\boldsymbol{\theta}}_m) = \operatorname*{argmin}_{\beta, \boldsymbol{\theta}} \sum_{i=1}^{n} L(y_i, f^{(m-1)}(\boldsymbol{x}_i) + \beta g(\boldsymbol{x}_i; \boldsymbol{\theta})), \quad \text{for } m = 1, \ldots, M,$$

rather than

$$\{(\hat{\beta}_m, \hat{\boldsymbol{\theta}}_m)\}_{m=1}^{M} = \operatorname*{argmin}_{\{(\beta'_m, \boldsymbol{\theta}'_m)\}_{m=1}^{M}} \sum_{i=1}^{n} L(y_i, \sum_{m=1}^{M} \beta'_m g(\boldsymbol{x}_i; \boldsymbol{\theta}'_m)).$$

Starting from an initial estimate of $f^{(0)}$, which usually is a constant, a base learner is fitted to the negative gradient of a loss function $L$, evaluated at the current estimate $f^{(0)}$. The resulting base learner is subsequently used to update the estimate $\hat{f}$ by adding it to $f^{(0)}$. This procedure is repeated until a stopping criterion or the maximum number of iterations is reached. Therefore, the final estimate $\hat{f}(\boldsymbol{x})$ is an ensemble of $M + 1$ base learners:

$$\hat{f}(\boldsymbol{x}) = \sum_{m=0}^{M} \beta_m g(\boldsymbol{x}; \theta_m). \tag{3.86}$$

## 3.4.2 Regularization

The complexity of the final ensemble is determined by 1) the total number of iterations $M$, which corresponds to its size, and 2) the learning rate $\nu$. The choice of $M$ and

**Algorithm 3.3:** Gradient boosting with Shrinkage.

```
Input: Differentiable loss function $L$, training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i\}_{i=1}^n$, number of steps
       $M > 0$, learning rate $0 < \nu \leq 1$.
Output: Additive ensemble of base learners $\hat{f}(\boldsymbol{x}) = \hat{f}^{(0)}(\boldsymbol{x}) + \sum_{m=1}^M \nu \rho_m g(\boldsymbol{x}; \boldsymbol{\theta}_m)$.
1 Initialize first base learner:
      $\hat{f}^{(0)}(\boldsymbol{x}) = \operatorname*{argmin}_{u \in \mathbb{R}} \sum_{i=1}^N L(y_i, u).$
2 for $m \leftarrow 1$ to $M$ do
3     Compute negative gradient and evaluate it at $\hat{f}^{(m-1)}(\boldsymbol{x})$ to obtain "pseudoresponses" $\tilde{y}_i$:
          $\tilde{y}_i = - \left[ \frac{\partial L(y_i, f(\boldsymbol{x}_i))}{\partial f(\boldsymbol{x}_i)} \right]_{f(\boldsymbol{x}) = \hat{f}^{(m-1)}(\boldsymbol{x})}, \quad \forall i = 1, \ldots, n.$
4     Fit base learner $g(\cdot)$ to the "pseudoresponses":
          $(\hat{a}_m, \hat{\boldsymbol{\theta}}_m) = \operatorname*{argmin}_{a, \boldsymbol{\theta}} \sum_{i=1}^n (\tilde{y}_i - a g(\boldsymbol{x}_i; \boldsymbol{\theta}))^2 .$
5     Determine step size $\rho > 0$ by line search:
          $\rho_m = \operatorname*{argmin}_{\rho} \sum_{i=1}^n L(y_i, \hat{f}^{(m-1)}(\boldsymbol{x}_i) + \rho g(\boldsymbol{x}_i; \hat{\boldsymbol{\theta}}_m)).$
6     Update approximation: $\hat{f}^{(m)}(\boldsymbol{x}) = \hat{f}^{(m-1)}(\boldsymbol{x}) + \nu \rho_m g(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_m))$.
7 end
```

$\nu$ has considerable impact on the generalization performance of the ensemble due to the well known bias-variance tradeoff: with increasing model complexity the bias towards the training data increases while the variance on unseen data increases (see e.g. [131]). In particular without shrinkage ($\nu = 1$), Friedman [104] observed that overfitting already occurs after relatively small number of iterations; Friedman suggested $\nu \in \{0.25, 0.125, 0.06\}$. Moreover, $\nu$ and $M$ are strongly connected, choosing smaller values for $\nu$ suggests increasing the value for $M$. Optimal values are usually determined by cross-validation.

In addition to shrinkage, stochastic gradient boosting [105] can be employed to remedy overfitting. Stochastic gradient boosting [105] is based on injecting randomness into training of the ensemble similar to another popular ensemble method: bootstrap aggregation or "bagging" [35, 36]. Instead of fitting each base learner to the whole training data consisting of $n$ samples, each gradient descent step is based on a different subset of the training data (see line 4 of algorithm 3.3). As in bagging [35], samples that have not been used in the current iteration can be used to obtain an out-of-bag error, which can be used to monitor convergence. For each iteration, Friedman [105] suggested drawing a subsample without replacement of approximately half the size of the full training data, because of its similarity to drawing a bootstrap sample (with replacement) [44, 103]. As a side effect, the time needed for each iteration is lowered due to subsampling as well.

Rashmi and Gilad-Bachrach [238] proposed another alternative to shrinkage, inspired by the success of dropout, which is used to regularize neural networks by randomly removing units during training [15, 144]. Whereas traditional gradient boosting considers all previously fitted base learners at each gradient descent step, Rashmi and Gilad-Bachrach [238] proposed to drop a randomly selected set of base learners at each iteration. The updated gradient boosting algorithm using dropout instead of shrinkage is depicted in algorithm 3.4.

Line 7 ensures that there always is at least one base learner that is dropped. Care has to be taken when adding a new base learner to the ensemble. If in one iteration $K$ base learners are dropped, a new base learner will try to compensate for the dropped base learners too. Therefore, its influence is going to be $K$ times higher than each of the dropped base learners, which can be corrected by assigning it the weight $K^{-1}$. Whereas in original gradient boosting (without shrinkage) the contribution of each base learner is only determined once via line search (see line 5 in algorithm 3.3), the contribution of dropped base learners has to be adjusted when using dropout, because they overlap with the new base learner. Formally, in the $m$-th iteration the contribution of all dropped base learners and the newly added base learner should be scaled by a factor $\eta$ such that the following equality holds:

$$\sum_{k \in \mathcal{D}_m} \rho_k = \eta \left( \rho_m + \sum_{k \in \mathcal{D}_m} \rho_k \right) , \tag{3.87}$$

which yields $\eta = \left(\sum_{k \in \mathcal{D}_m} \rho_k\right) / \left(\rho_m + \sum_{k \in \mathcal{D}_m} \rho_k\right)$. If no line search is performed, $\rho_m = 1$ and $\rho_k = 1 \; \forall k \in \mathcal{D}_m$ and the scaling factor simplifies to $\eta = |\mathcal{D}_m| / (1 + |\mathcal{D}_m|)$, which is the case in algorithm 3.3. Finally, dropout can also be combined with stochastic gradient boosting.

### 3.4.3 Common Base Learners

The choice of a base learner is not pre-determined in gradient boosting and constitutes another hyper-parameter. Due to the greedy stagewise nature of gradient boosting, it is usually preferred to employ simple models that are fast to train, because errors can be compensated for by base learners in later iterations. Common choices include regression trees [39] and componentwise least squares [45].

**Algorithm 3.4:** Gradient boosting with dropout.

```
Input: Differentiable loss function $L$, training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i\}_{i=1}^{n}$, number of steps
        $M > 0$, dropout rate $0 < \varepsilon < 1$.
Output: Additive ensemble of base learners $\hat{f}(\boldsymbol{x})$.
Initialize first base learner:
    $$\hat{f}^{(0)}(\boldsymbol{x}) = \operatorname*{argmin}_{u \in \mathbb{R}} \sum_{i=1}^{N} L(y_i, u).$$
$\boldsymbol{b} = \mathbb{1}_m$
for $m \leftarrow 1$ to $M$ do
    if $m > 1$ then
        $\mathcal{D}_m \leftarrow$ Randomly selected subset of $m - 1$ base learners, such that each previously
        constructed base learner is included with probability $\varepsilon$.
        if $\mathcal{D}_m = \varnothing$ then
            $\mathcal{D}_m \leftarrow$ Randomly drawn integer from $[1; m - 1]$.
        end
    else
        $\mathcal{D}_m \leftarrow \varnothing$
    end
    Drop base learners in $\mathcal{D}_m$ from ensemble, yielding model $\hat{g}(\boldsymbol{x})$:
        $$\hat{g}(\boldsymbol{x}) \leftarrow \sum_{\substack{k=1 \\ k \notin \mathcal{D}_m}}^{m-1} b_k \cdot g(\boldsymbol{x}; \boldsymbol{\theta}_k).$$
    Compute negative gradient and evaluate it at $\hat{g}(\boldsymbol{x})$ to obtain "pseudoresponses" $\tilde{y}_i$:
        $$\tilde{y}_i = -\left[\frac{\partial L(y_i, f(\boldsymbol{x}_i))}{\partial f(\boldsymbol{x}_i)}\right]_{f(\boldsymbol{x}) = \hat{g}(\boldsymbol{x})}, \quad \forall i = 1, \ldots, n.$$
    Fit base learner $g(\cdot)$ to the "pseudoresponses":
        $$(\hat{a}_m, \hat{\boldsymbol{\theta}}_m) = \operatorname*{argmin}_{a, \boldsymbol{\theta}} \sum_{i=1}^{n} (\tilde{y}_i - a g(\boldsymbol{x}_i; \boldsymbol{\theta}))^2 .$$
    $b_m \leftarrow 1/(|D| + 1)$                                    /* Store scaling factor */
    foreach $k \in \mathcal{D}_m$ do
        $b_k \leftarrow b_k \cdot \frac{|\mathcal{D}_m|}{|\mathcal{D}_m| + 1}$          /* Update scaling factor of dropped base learner */
    end
end
$\hat{f}(\boldsymbol{x}) = \hat{f}^{(0)}(\boldsymbol{x}) + \sum_{m=1}^{M} b_m g(\boldsymbol{x}; \boldsymbol{\theta}_m)$
```

### Regression Trees

A major advantage of regression trees is that they can naturally deal with data consisting of features with varying statistical properties (continuous, ordinal, or categorical). To speed up training, they are often randomized [37] or restricted to just two leaf nodes (so called "stumps"). A "stump" splits the data based on a single variable only and does not consider interactions between features, whereas a tree of $d$ levels considers interactions up to order $d-1$. Typically, the range of $d$ is within $[4;8]$ and can be determined via cross-validation [129]. A regression tree with axis-aligned splits is built in a top-down greedy manner by splitting the data into two subsets according to split point $\tau$ of variable $j$ such that the squared error in the two subsets is minimized:

$$\min_{j,\tau}\left[\min_{c_{\text{left}}}\sum_{i\in\{k|x_{kj}\leq\tau\}}\left(\tilde{y}_i-c_{\text{left}}\right)^2+\min_{c_{\text{right}}}\sum_{i\in\{k|x_{kj}>\tau\}}\left(\tilde{y}_i-c_{\text{right}}\right)^2\right]. \tag{3.88}$$

This process is repeated recursively until the desired depth of the tree is reached or the number of samples reaching a node is too low to justify splitting the data again. Each leaf is associated with a prediction model, which is constructed from samples that reached that leaf; in the simplest case, it is just the average "pseudoresponse" $\tilde{y}_i$ of all samples in the leaf. During prediction, a new data point is guided through the tree according to the split criterion in the inner nodes until a leaf node is reached and the corresponding leaf model is applied. Therefore, a regression tree with $J$ leaves itself is an additive model of the form

$$g(\boldsymbol{x};\boldsymbol{\theta}_m)=\sum_{j=1}^{J}\boldsymbol{\theta}_{mj}I(\boldsymbol{x}\in R_{mj}),$$

where $\boldsymbol{\theta}_{mj}$ denotes the parameters of $j$-th leaf model and the indicator function denotes whether $\boldsymbol{x}$ reached the $j$-th leaf node. Consequently, when using regression trees, the ensemble $\hat{f}(\boldsymbol{x})$ is in fact extended by $J$ base learners in each gradient descent step.

### Componentwise Least Squares

Componentwise least squares [45] is based on ordinary linear least squares with only a single feature. At each gradient descent step, the feature that reduces the squared error with respect to the "pseudoresponse" the most is selected. Thus, fitting a base learner in line 4 of algorithm 3.3 comes down to solving

$$j_m^*=\operatorname*{argmin}_{j=1,\ldots,p}\sum_{i=1}^{n}\left(\tilde{y}_i-\hat{\alpha}_jx_{ij}\right)^2, \tag{3.89}$$

where $\hat{\alpha}_j$ denotes the ordinary least squares solution

$$\hat{\alpha}_j=\frac{\sum_{i=1}^{n}x_{ij}\tilde{y}_i}{\sum_{i=1}^{n}x_{ij}^2}. \tag{3.90}$$

The parameters of the base learner in the $m$-th iteration are a vector with $p$ components that is all zero except at position $j_m^*$:

$$\hat{\boldsymbol{\theta}}_m = \left(0, \ldots, 0, \hat{\alpha}_{j_m^*}, 0, \ldots, 0\right)^\top .$$

The resulting ensemble after $M$ iterations is then given by

$$\hat{f}(\boldsymbol{x}) = \sum_{m=1}^{M} \nu \cdot \hat{\alpha}_{j_m^*} x_{j_m^*} = \sum_{m=1}^{M} \nu \cdot \boldsymbol{x}^\top \hat{\boldsymbol{\theta}}_m = \boldsymbol{x}^\top \left( \sum_{m=1}^{M} \nu \cdot \hat{\boldsymbol{\theta}}_m \right) . \tag{3.91}$$

Starting with an initial model $\boldsymbol{\theta}_0$, usually of all zeros, only a single coefficient in the vector $\boldsymbol{\theta} \in \mathbb{R}^p$ is updated in each step. In particular, if the loss function $L$ is the squared error loss, the solution of gradient boosting with infinitesimally learning rate $\nu$ approximates the LASSO solution [287], thus resulting in a model that has feature selection embedded [43, 83, 129]. Because of its similarity to the LASSO, using componentwise least squares as a base learner is especially suitable when dealing with high-dimensional data [42].

Note that the linear model in eq. (3.91) does not explicitly contain an intercept term, but it can be easily extended by adding an additional feature of all ones to accommodate the intercept. Finally, componentwise smoothing spline follow a similar concept as componentwise least squares but add more flexibility due to fitting a smoothed curve to individual features rather than a linear model [45].

### Likelihood-based Boosting

Likelihood-based boosting [290, 291] can be used to fit generalized additive models and generalized linear models via gradient boosting. The objective in likelihood-based boosting is to maximize the log-likelihood function $LL(\boldsymbol{\theta})$ of a generalized linear model, which includes, among others, least squares, logistic regression, and Cox's proportional hazards model [67]. Instead of repeatedly fitting a base learner to the negative gradient of a loss function, evaluated at the previous iteration's estimate, likelihood-based boosting takes a single Newton step in each iteration and accounts for previous iterations by introducing an offset variable into the log-likelihood function. *Partial likelihood-based boosting* behaves similar to componentwise least squares in the sense that only a single coefficient is updated via one Newton step in each iteration, which leads to sparse solutions [291]. Tutz and Binder [291] showed that their proposed procedure is not limited to merely updating a single feature in each iteration but can also be employed to force updates of certain groups of features, such as mandatory features or features that encode a single categorical variable. Algorithm 3.5 presents partial likelihood-based boosting, which I will detail next.

The main principle of algorithm 3.5 is that only a single coefficient is updated in each iteration. It is important to point out that the core of the method is line 4,

**Algorithm 3.5:** Partial likelihood-based boosting.

```
Input: Penalized, restricted log-likelihood $LL^j_\lambda$, training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i\}_{i=1}^n$, number of
       steps $M > 0$.
Output: Additive ensemble of base learners $\hat{f}(\boldsymbol{x}) = \boldsymbol{x}^\top \hat{\boldsymbol{\theta}}_M$.
1  Initialize $\hat{\boldsymbol{\theta}}_0 = \mathbf{0}_p$ and $\hat{\boldsymbol{\eta}}_0 = \mathbf{0}_n$.
2  for $m \leftarrow 1$ to $M$ do
3      for $j \leftarrow 1$ to $p$ do                                  /* Iterate over features */
4          Compute gradient $\boldsymbol{g}_j$ and Hessian $\boldsymbol{H}_j$ of penalized log-likelihood restricted to the
           $j$-th feature $LL^j_\lambda$ and evaluate them at $\boldsymbol{\theta} = \mathbf{0}_p$ and offsets $\hat{\boldsymbol{\eta}}_{m-1}$ of the previous
           iteration:
           $$\boldsymbol{g}_j = \left[\frac{\partial LL^j_\lambda(\boldsymbol{\theta}; \hat{\boldsymbol{\eta}}_{m-1})}{\partial \theta_j}\right]_{\boldsymbol{\theta}=\mathbf{0}_p}, \qquad \boldsymbol{H}_j = \left[\frac{\partial^2 LL^j_\lambda(\boldsymbol{\theta}; \hat{\boldsymbol{\eta}}_{m-1})}{\partial \theta_j^2}\right]_{\boldsymbol{\theta}=\mathbf{0}_p}.$$
5          Perform one Newton step to obtain update for the $j$-th coefficient:
           $$\hat{\alpha}_j = -\left(\boldsymbol{H}_j\right)^{-1} \boldsymbol{g}_j.$$
           $\hat{\boldsymbol{u}}_j = (0, \ldots, 0, \hat{\alpha}_j, 0, \ldots, 0)^\top$
6      end
7      $j^* = \operatorname{argmin}_{j=1,\ldots,p} -2 \cdot LL^j_\lambda(\hat{\boldsymbol{u}}_j; \hat{\boldsymbol{\eta}}_{m-1})$          /* Select the best update */
8      $\hat{\boldsymbol{\theta}}_m = \hat{\boldsymbol{\theta}}_{m-1} + \hat{\boldsymbol{u}}_{j^*}$                                  /* Update ensemble */
9      $\hat{\boldsymbol{\eta}}_m = \hat{\boldsymbol{\eta}}_{m-1} + \boldsymbol{X}\hat{\boldsymbol{u}}_{j^*}$                               /* Update offset */
10 end
```

where gradient and Hessian of a penalized log-likelihood function, restricted to the $j$-th coefficient, are computed. The penalized, restricted log-likelihood function has the form

$$LL^j_\lambda(\boldsymbol{\theta}; \boldsymbol{\eta}) = LL^j(\theta_j; \boldsymbol{\eta}) - \frac{\lambda}{2} \boldsymbol{\theta}^\top \boldsymbol{P} \boldsymbol{\theta}, \tag{3.92}$$

where $LL^j(\theta_j; \boldsymbol{\eta})$ denotes the restricted likelihood function that dropped all but the $j$-th feature, $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)^\top \in \mathbb{R}^n$ denotes a vector of offset terms, $\boldsymbol{P} \in \mathbb{R}^{p \times p}$ a structured penalty (usually the identity), and $\lambda > 0$ a regularization parameter. Each possible update is in effect obtained via a single Newton step starting from the estimate $\boldsymbol{\theta} = \mathbf{0}_p$ (see line 5 of algorithm 3.5). Taking a single Newton step is sufficient because each iteration only requires fitting a weak learner, and starting the Newton step from the origin is possible because the offset already accounts for previous updates. After computing all possible updates, the one maximizing the improvement in deviance is selected and applied to the coefficients of the additive model. Note that likelihood-based boosting always applies the full update and does not shrink it as gradient boosting in algorithm 3.3 does. Instead, the regularization parameter $\lambda$ controls the size of the updates and should be chosen large enough such that the update in each iteration is small [28].

## 3.4.4 Boosting Methods for Survival Analysis

In this section, I will describe models for survival analysis that optimize their particular loss function via gradient boosting. The main difference between the models presented in this section is the choice of loss function. Some authors perform boosting based on the negative log partial likelihood of Cox's proportional hazards model [28, 193, 239], based on the accelerated failure time model [148, 253, 310], or they derive a loss function from evaluation measures [25, 211].

### Based on Cox's Proportional Hazards Model

The earliest approach by Ridgeway [239] used boosting to construct an additive model that maximizes the log partial likelihood function of Cox's proportional hazards model [67]; thereby replacing the linear model in eq. (3.29) with an additive model $f(\boldsymbol{x})$:

$$\log PL(f) = \sum_{i=1}^{n} \delta_i \left[ f(\boldsymbol{x}_i) - \log \left( \sum_{j \in \mathcal{R}_i} \exp(f(\boldsymbol{x}_j)) \right) \right] . \tag{3.93}$$

Equation (3.93) is maximized with respect to the model $f$, which is performed in function space via gradient boosting following algorithm 3.3. The gradient with respect to $f(\boldsymbol{x}_k)$ is given by

$$\frac{\partial \log PL(f)}{\partial f(\boldsymbol{x}_k)} = \delta_k - \sum_{i=1}^{n} \delta_i \frac{I(y_k \geq y_i) \exp(f(\boldsymbol{x}_k))}{\sum_{j \in \mathcal{R}_i} \exp(f(\boldsymbol{x}_j))} . \tag{3.94}$$

In the $m$-th iteration, a base learner is fit to the "pseudoresponses" $\tilde{y}_k$ corresponding to the negative gradient of the negative log partial likelihood evaluated at the previous estimate $\hat{f}^{(m-1)}$:

$$\tilde{y}_k = \delta_k - \sum_{i=1}^{n} \delta_i \frac{I(y_k \geq y_i) \exp(\hat{f}^{(m-1)}(\boldsymbol{x}_k))}{\sum_{j \in \mathcal{R}_i} \exp(\hat{f}^{(m-1)}(\boldsymbol{x}_j))}, \qquad \forall k = 1, \ldots, n. \tag{3.95}$$

Ridgeway [239] chose regression "stumps" as base learners during optimization. Li and Luan [193] follow the same approach as Ridgeway [239], but used componentwise cubic smoothing splines as base learners [45].

Binder and Schumacher [28] optimized the loss function (3.93) too, but performed optimization via likelihood-based boosting instead (see algorithm 3.5). To account for the offset terms $\boldsymbol{\eta}$ in the log partial likelihood function, they used a modified unpenalized log partial likelihood function, which accommodates the offset term and is restricted to the $j$-th feature:

$$LL^j(\theta_j; \boldsymbol{\eta}) = \sum_{i=1}^{n} \delta_i \left[ \eta_i + x_{ij}\theta_j - \log \left( \sum_{k \in \mathcal{R}_i} \exp(\eta_k + x_{kj}\theta_j) \right) \right]. \tag{3.96}$$

The first- and second-order partial derivatives of $LL^j(\theta_j; \boldsymbol{\eta})$ with respect to the $j$-th coefficient are

$$\begin{aligned} \frac{\partial LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j} &= \sum_{i=1}^{n} \delta_i \left[ x_{ij} - \frac{\sum_{k \in \mathcal{R}_i} \exp(\eta_k + x_{kj}\theta_j) x_{kj}}{\sum_{k \in \mathcal{R}_i} \exp(\eta_k + x_{kj}\theta_j)} \right] \\ \frac{\partial^2 LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j^2} &= -\sum_{i=1}^{n} \delta_i \left[ \frac{\sum_{k \in \mathcal{R}_i} \exp(\eta_k + x_{kj}\theta_j) x_{kj}^2}{\sum_{k \in \mathcal{R}_i} \exp(\eta_k + x_{kj}\theta_j)} - \left( \frac{\sum_{k \in \mathcal{R}_i} \exp(\eta_k + x_{kj}\theta_j) x_{kj}}{\sum_{k \in \mathcal{R}_i} \exp(\eta_k + x_{kj}\theta_j)} \right)^2 \right]. \end{aligned}$$

Finally, the update for the $j$-th coefficient (see line 5 of algorithm 3.5) can be computed by

$$\hat{\alpha}_j = -\left( \frac{\partial^2 LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j^2} - \lambda \right)^{-1} \frac{\partial LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j}, \tag{3.97}$$

where $\theta_j$ is evaluated at the origin.

To select the best update in line 7 of algorithm 3.5, Binder and Schumacher [28] used a modified score statistic based on the first-order Taylor approximation of the penalized log-likelihood function about 0, which yields

$$\begin{aligned} LL^j_\lambda(\hat{\boldsymbol{u}}_j; \boldsymbol{\eta}) &\approx LL^j(\mathbf{0}_p; \boldsymbol{\eta}) + \frac{\partial LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j} (\hat{\alpha}_j - 0) \\ &= LL^k(\mathbf{0}_p; \boldsymbol{\eta}) + \frac{\partial LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j} \left( -\frac{\partial^2 LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j^2} + \lambda \right)^{-1} \frac{\partial LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j}, \end{aligned}$$

where $\hat{\alpha}_j$ is the $j$-th element of $\hat{\boldsymbol{u}}_j$ defined in (3.97) and gradient and Hessian are evaluated at $\theta_j = 0$. To determine the best update, the constant first term can be ignored and the remaining quantities are already available from computing the Newton step, which avoids additional computational costs to compute the full penalized log-likelihood function, especially for high-dimensional data. Line 7 of algorithm 3.5 becomes

$$j^* = \underset{j=1,\ldots,p}{\operatorname{argmin}} \left( \frac{\partial LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j} \right)^2 \left( \frac{\partial^2 LL^j(\theta_j; \boldsymbol{\eta})}{\partial \theta_j^2} - \lambda \right)^{-1}. \tag{3.98}$$

### Based on Accelerated Failure Time Model

Hothorn *et al.* [148] extended boosting of the squared error loss to censored data by introducing inverse probability of censoring weights into the fitting process of base learners. Each sample $i$ is associated with a weight $\omega_i$ proportional to the inverse probability of being censored after time $y_i$, given its set of features $\boldsymbol{x}_i$ (see section 3.1.2 on page 34). By including inverse probability of censoring weights into the optimization step in line 4 of algorithm 3.3, one obtains

$$(\hat{a}_m, \hat{\boldsymbol{\theta}}_m) = \underset{a,\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{n} \omega_i \left( \tilde{y}_i - a g(\boldsymbol{x}_i; \boldsymbol{\theta}) \right)^2 . \tag{3.99}$$

They assumed that censoring is independent of the features and used the non-parametric Kaplan-Meier estimator (2.19) to compute the weights. Moreover, they employed componentwise weighted least squares [45] as base learner, but any base learner that takes into account sample weights could be used.

Wang and Wang [310] optimized the squared error loss as well, but accounted for censoring using the imputation approach of Buckley and James [41] (cf. section 3.1.1 on page 33). In the $m$-th gradient descent iteration, the survival times of censored patients are imputed based on the estimate $\hat{f}^{(m-1)}$ of the previous iteration and the corresponding residuals $r_i = y_i - \hat{f}^{(m-1)}(\boldsymbol{x}_i)$. Imputed survival times $\tilde{t}_i^{(m)}$ are computed by

$$\tilde{t}_i^{(m)} = \delta_i t_i + (1 - \delta_i) \left[ \hat{f}^{(m-1)}(\boldsymbol{x}_i) + \sum_{k=1}^{n} \delta_k w_{ik} r_k \right] , \tag{3.100}$$

with

$$w_{ik} = \begin{cases} \Delta \hat{S}(r_k) [\hat{S}(r_i)]^{-1} & \text{if } r_i < r_k, \\ 0 & \text{else.} \end{cases} \tag{3.101}$$

Next, a base learner is fit to the negative gradient of the squared error loss with respect to the imputed survival times $\tilde{t}_i^{(m)}$ and then the estimate $\hat{f}(\boldsymbol{x})$ is updated. These two steps are repeated until convergence or estimates oscillate due to discontinuities in the step function obtained by the Kaplan-Meier estimator [167]. Wang and Wang [310] experimented with componentwise least squares, componentwise smoothing splines, and regression trees as base learners and compared their approach against the boosting method in [148] as well as penalized least squares methods [152, 308].

In contrast to the previous two approaches, Schmid and Hothorn [253] applied gradient boosting to the fully parametric accelerated failure time model, for which the baseline survival function needs to be specified. The advantage is that optimization can be performed via maximum likelihood.

#### Based on Evaluation Measure

Finally, I will present two approaches that are not based on existing linear models for survival analysis, but on performance measures that usually assess a model's fit or predictive capabilities. The loss function optimized by Benner [25] is based on the

Brier score for censored data (see [117] and section 3.7.3 on page 89). They estimated an additive model $f(\boldsymbol{x})$ to predict the probability of remaining event-free up to time $t$ by minimizing the following loss function

$$L^t_{\text{Brier}}(f) = \sum_{i=1}^n I(y_i \leq t \wedge \delta_i = 1) \frac{(0 - f(\boldsymbol{x}_i))^2}{\hat{G}(y_i)} + I(\delta_i > t) \frac{(1 - f(\boldsymbol{x}_i))^2}{\hat{G}(t)}, \tag{3.102}$$

where $\hat{G}(c)$ denotes the Kaplan-Meier estimator (2.19) of $P(C > c)$, i.e., the probability of being censored after time point $c$. The gradient of the loss function $L^t_{\text{Brier}}$ with respect to $f(\boldsymbol{x}_k)$ is given by

$$\frac{\partial L^t_{\text{Brier}}(f)}{\partial f(\boldsymbol{x}_k)} = 2 \left( I(y_k \leq t \wedge \delta_k = 1) \frac{0 - f(\boldsymbol{x}_k)}{\hat{G}(y_k)} + I(\delta_k > t) \frac{1 - f(\boldsymbol{x}_k)}{\hat{G}(t)} \right). \tag{3.103}$$

Benner [25] constructed their ensemble by fitting regression trees to the "pseudoresponses."

Mayr and Schmid [211] proposed to apply gradient boosting to a smoothed version of the concordance index by Uno *et al.* [292] (cf. section 3.7.1 on page 84). They maximized the following loss function:

$$\begin{aligned} L_{\text{CI}}(f) &= \sum_{i=1}^n \sum_{j=1}^n w_{ij} \left[ 1 + \exp\left( \frac{f(\boldsymbol{x}_j) - f(\boldsymbol{x}_i)}{\sigma} \right) \right]^{-1}, \\ w_{ij} &= \frac{\omega_i^2 I(y_i < y_j)}{\sum_{u=1}^n \sum_{v=1}^n \omega_u^2 I(y_u < y_v)}, \end{aligned} \tag{3.104}$$

where $\omega_i$ are inverse probability of censoring (IPC) weights (see section 3.1.2 on page 34), and $\sigma > 0$ is a hyper-parameter that controls the smoothness of the approximation of the indicator function. The gradient of the loss function can be derived as

$$\begin{aligned} \frac{\partial L_{\text{CI}}(f)}{\partial f(\boldsymbol{x}_k)} = \sum_{i=1}^n & \frac{w_{ki} \exp((f(\boldsymbol{x}_i) - f(\boldsymbol{x}_k))/\sigma)}{\sigma \left(1 + \exp((f(\boldsymbol{x}_i) - f(\boldsymbol{x}_k))/\sigma)\right)^2} \\ & - \frac{w_{ik} \exp((f(\boldsymbol{x}_k) - f(\boldsymbol{x}_i))/\sigma)}{\sigma \left(1 + \exp((f(\boldsymbol{x}_k) - f(\boldsymbol{x}_i))/\sigma)\right)^2} \end{aligned} \tag{3.105}$$

They employed componentwise least squares as base learner and used the Kaplan-Meier estimator (2.19) to compute IPC weights.

## 3.5 Tree-based Methods for Survival Analysis

### 3.5.1 Recursive Partitioning

The survival models I described previously can all be trained by solving a – usually convex – optimization problem. If the relationship between features and survival time

**Algorithm 3.6:** Recursive training of tree-based methods with axis-aligned splits.

```
Function SplitNode($\mathcal{D}$)
    Input: Training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^n$.
    Output: Leaf models $\mathcal{M}$.
    if split is allowed then
        for $j = 1$ to $p$ do                                        /* Iterate over features */
            $\tau_j \leftarrow$ GetCutPoint($\{x_{ij}\}_{i=1}^n$)
            $\mathcal{I}_L^{(j)} \leftarrow \{i \mid x_{ij} \leq \tau_j\}_{i=1}^n$
            $\mathcal{I}_R^{(j)} \leftarrow \{i \mid x_{ij} > \tau_j\}_{i=1}^n$
            $\rho_j \leftarrow$ SplitCriterion($\mathcal{D}, \mathcal{I}_L^{(j)}, \mathcal{I}_R^{(j)}$)
        end
        $j^* = \operatorname{argmax}_{j=1,\ldots,n} \rho_j$                           /* Determine best split */
        $\mathcal{D}_L \leftarrow \{(\boldsymbol{x}_i, y_i, \delta_i) \mid i \in \mathcal{I}_L^{(j*)}\}$
        $\mathcal{D}_R \leftarrow \{(\boldsymbol{x}_i, y_i, \delta_i) \mid i \in \mathcal{I}_R^{(j*)}\}$
        $\mathcal{M} \leftarrow$ SplitNode($\mathcal{D}_L$) $\cup$ SplitNode($\mathcal{D}_R$)
    else
        $\mathcal{M} \leftarrow$ CreateLeafModel($\mathcal{D}$)
    end
    return Leaf models $\mathcal{M}$
end
```

is described by a linear model, the overall model and effects of features on survival can be easily interpreted. Although such models can also be used with more complex data that includes non-linear effects and interactions between features, these properties have to be manually defined by the analyst for the model to consider. Depending on the level of complexity, researchers might be forced to try many different model formulations, which is cumbersome. In these situations, tree-based methods often offer a better alternative because of their ability to automatically handle certain types of non-linearities and interactions. Tree-based methods for survival analysis are based on the seminal work of Breiman *et al.* [39] on Classification and Regression Trees (CART).

A tree-based survival model is described by a hierarchical binary tree consisting of internal nodes, which divide subjects into subgroups, and leaf nodes, which contain predictive models specific to the particular subgroup of patients that reached that node. As mentioned in section 3.4.3 (page 65) on regression trees, a tree model can also be interpreted as being additive in its leaf models. Tree-based methods have two properties that sets them apart from survival models I described above: 1) a tree can naturally handle features with different statistical properties (continuous, ordinal, categorical), and 2) a tree of depth $d$ automatically considers interactions up to order $d - 1$ [129].

When constructing a tree, the training data is split into $J$ disjoint partitions in a greedy top to bottom manner, resulting in $J$ leaf models (see algorithm 3.6). To avoid overfitting, it is often necessary to limit the depth of the tree or the number of leaf nodes. This can be achieved by explicitly specifying the extent of a tree or by defining the minimum number of samples that are sufficient to justify another split. From algorithm 3.6 it is clear that training is in essence determined by the choice of split criterion and leaf model, which I will describe next.

### 3.5.2 Split Criteria

Several authors proposed criteria for splitting nodes in a survival tree, which are based on determining the difference in survival in the left and right child node, represented by scalar $\rho$. Ultimately, the split maximizing $\rho$ is carried out and the same procedure is applied to the resulting child nodes. Table 3.2 provides an overview of the split criteria I will explain in more detail below.

Let $\mathcal{J} \subseteq \{1, \ldots, K\}$ indicate the set of $J = |\mathcal{J}|$ terminal nodes of a tree with $K$ nodes. The set $\mathcal{I}_k$ with $1 \leq k \leq K$ indicates the samples assigned to the $k$-th node. When splitting a parent node into two child nodes, sets $\mathcal{I}_L$ and $\mathcal{I}_R$ denote the samples that are assigned to the left and right child node; $n_L = |\mathcal{I}_L|$, $n_R = |\mathcal{I}_R|$, and $n_P = n_L + n_R$ indicate the number of samples in the left, right, and parent node, respectively. The super- or subscripts $P$, $L$, and $R$ indicate whether a function or value is computed with respect to samples in the parent node, left child node, and right child node, respectively.

The earliest split criterion was based on measuring the $L^1$-Wasserstein distance between Kaplan–Meier estimated survival functions $\hat{S}_L$ and $\hat{S}_R$ of child nodes [116]. It represents the area between estimated survival functions $\hat{S}_L$ and $\hat{S}_R$, and is computed by

$$\rho_j = \int_0^{t_{\max}} |\hat{S}_L(t) - \hat{S}_R(t)| dt, \tag{3.106}$$

with $t_{\max} = \min(\max\{y_i | \delta_i = 1\}_{i=1}^{n_L}, \max\{y_i | \delta_i = 1\}_{i=1}^{n_R})$ being the maximum event time for which both $\hat{S}_{\mathrm{L}}$ and $\hat{S}_{\mathrm{R}}$ are defined.

When the objective is to determine whether survival functions of child nodes differ from each other, it is natural to use any of the statistical tests presented in section 2.5 on page 25. LeBlanc and Crowley [189] used the log-rank test statistic [207], Segal [261] the test statistic by Tarone and Ware [284], and Ciampi *et al.* [57] the test statistic by Gehan [108]. In each case, the quality of a split corresponds to the value of the test statistic $X^2$ in eq. (2.33) computed with respect to the corresponding weight function in table 2.1. To ease computation when computing the log-rank test, LeBlanc and

**Table 3.2**: Overview of split criteria to construct tree-based models for survival analysis.

| References | Split Criteria |
|---|---|
| Gordon and Olshen [116] | Maximum $L^1$-Wasserstein distance between Kaplan–Meier estimated survival functions. |
| LeBlanc and Crowley [189] | Max. log-rank test statistic (see section 2.5). |
| Ishwaran and Kogalur [157] and LeBlanc and Crowley [189] | Max. approximated log-rank test statistic. |
| Segal [261] | Max. extended log-rank test statistic by Tarone and Ware (see section 2.5). |
| Ciampi *et al.* [57] | Max. extended log-rank test statistic by Gehan (see section 2.5). |
| Hothorn and Lausen [149] | Min. $p$-value of maximally selected log-rank score statistic. |
| Keleş and Segal [171] and Therneau *et al.* [286] | Max. improvement in within-node homogeneity based on martingale or deviance residuals of null Cox model. |
| Davis and Anderson [71] | Max. likelihood of exponential model with constant hazard rate in child nodes. |
| LeBlanc and Crowley [188] | Max. reduction in deviance of a full likelihood Cox model. |
| Zhang [329] | Max. reduction in impurity with respect to observed times and event indicators in child nodes. |
| Ishwaran and Kogalur [157] and Ishwaran *et al.* [159] | Split that preserves conversation-of-events principle the most. |
| Jin *et al.* [162] | Max. difference in restricted mean survival times between child nodes. |
| Schmid *et al.* [255] | Max. Harrell's concordance index. |
| Cho and Hong [56] | Max. reduction in absolute distance to within-node median survival time. |
| Molinaro *et al.* [214] | Max. reduction in within-node variance with respect to IPC weighted survival times. |

Crowley [189] proposed to approximate the numerator of the log-rank test statistic (2.28) by

$$\left( \sum_{i=1}^{m_P} d_{Li} - r_{Li} \frac{d_{Li} + d_{Ri}}{r_{Li} + r_{Ri}} \right)^2 = \left( \sum_{i=1}^{m_P} d_{Li} - \sum_{k=1}^{n_P} I(x_{kj} \leq \tau) \hat{H}_P(y_k) \right)^2 , \tag{3.107}$$

where $m_P$ denotes the number of distinct event times in the parent node, $d_{Li}$ and $r_{Li}$ the number of events and patients at risk at time $t_i$ in the left child node, respectively, and $\hat{H}_P(\cdot)$ the Nelson-Aalen estimator (2.20) with respect to samples in the parent node. Variables $d_{Ri}$ and $r_{Ri}$ are defined analogous with respect to samples in the right child node. This approximation was further refined by Ishwaran and Kogalur [157] leading to the following approximation of the log-rank statistic

$$X^2 = \frac{\left(\sum_{i=1}^{m_P} d_{Li} + d_{Ri}\right) \left(\sum_{i=1}^{m_P} d_{Li} - \sum_{k=1}^{n_P} I(x_{kj} \leq \tau) \hat{H}_P(y_k)\right)^2}{\left(\sum_{k=1}^{n_P} I(x_{kj} \leq \tau) \hat{H}_P(y_k)\right) \left(\sum_{i=1}^{m_P} (d_{Li} + d_{Ri}) - \sum_{k=1}^{n_P} I(x_{kj} \leq \tau) \hat{H}_P(y_k)\right)}. \tag{3.108}$$

The advantage is that it is sufficient to compute the Nelson-Aalen estimator $\hat{H}_P(\cdot)$ once for the parent node when evaluating multiple possible splits.

Hothorn and Lausen [149] formulated the search for the best split as a test with the null hypothesis

$$H_0 : F(t_i \mid x_{ij} \leq \tau) = F(t_i \mid x_{ij} > \tau), \quad \forall t, \tau \in \mathbb{R}, 1 \leq j \leq p,$$

against the alternative that there is at least one feature $j$ and one cut point $\tau$ that results in two distinguishable subgroups. If for each combination of cut point and feature a two-sample linear rank statistic $S_{j,\tau}$ is computed, the objective amounts to

$$\max_{j=1,\ldots,p} \max_{\tau} |S_{j,\tau}|. \tag{3.109}$$

They showed that, under the null hypothesis, the maximally selected rank statistic (3.109) follows a multivariate normal distribution that depends on the correlation between two-sample linear rank statistics. In a last step, they adjusted the $p$-value for the number of possible splits and the different scales of features. The split with the lowest adjusted $p$-value is selected. Let $\gamma_k = \sum_{i=1}^{n_P} I(y_i \leq y_k)$ be the number of samples that are censored or experienced an event up to time $y_k$. The standardized test statistic $S_{j,\tau}$ can be computed from log-rank scores $a_i$ by

$$\begin{aligned} a_i &= \delta_i - \sum_{k=1}^{\gamma_i} \frac{\delta_j}{n_P - \gamma_k + 1}, \quad \forall i = 1, \ldots, n_P, \\ S_{j,\tau} &= \frac{\sum_{i=1}^{n_P} I(x_{ij} \leq \tau)(a_i - n_L \mu_a(P))}{\sqrt{n_L \left(1 - \frac{n_L}{n_P}\right) \sigma_a^2(P)}}, \end{aligned}$$

where $\mu_a(P)$ and $\sigma_a^2(P)$ denote the sample mean and sample variance of all log-rank scores of samples in the parent node [149, 157]. The combination of feature and threshold that maximizes $|S_{j,\tau}|$ is selected.

Therneau *et al.* [286] first suggested deriving martingale residuals from a null Cox model – one that does not consider any features – and use them to build a regression tree. Later, Keleş and Segal [171] formalized the idea, leading to the split criterion

$$\frac{n_L n_R}{n_P} \left( \frac{1}{n_L} \sum_{i \in \mathcal{I}_L} r(i) - \frac{1}{n_R} \sum_{i \in \mathcal{I}_R} r(i) \right)^2, \tag{3.110}$$

where $r(i)$ are either martingale or deviance residuals from a null Cox model:

$$r_{\mathrm{MR}}(i) = \delta_i - \hat{H}_P(y_i), \tag{3.111}$$

$$r_{\mathrm{DEV}}(i) = \operatorname{sign}(r_{\mathrm{MR}}(i))\sqrt{-2(r_{\mathrm{MR}}(i) + \delta_i \log(\delta_i - r_{\mathrm{MR}}(i)))}. \tag{3.112}$$

Davis and Anderson [71] proposed a split criterion based on the log-likelihood of an exponential model with constant hazard rate. The log-likelihood of the $k$-th node is defined as

$$LL(k) = \left( \sum_{i \in \mathcal{I}_k} \delta_i \right) - \left( \sum_{i \in \mathcal{I}_k} \delta_i \right) \log \left( \frac{\sum_{i \in \mathcal{I}_k} \delta_i}{\sum_{i \in \mathcal{I}_k} y_i} \right). \tag{3.113}$$

Care has to be taken when a node only contains censored samples, which would result in $\log 0$; instead, the logarithm is replaced by $\log(0.5 \sum_{i \in \mathcal{I}_k} y_i)$. The split that maximizes the log-likelihood in both child nodes is selected.

The split criterion by LeBlanc and Crowley [188] selects the split with the greatest reduction in deviance of a full likelihood Cox model (see section 3.2.2 on page 37). Instead of using the exponential of a linear model, each terminal node $k \in \mathcal{J}$ contains its specific Cox model with hazard function $h(t) = \theta_k h_0(t)$, where $\theta_k > 0$ is the constant model of the $k$-th node that represent the relative risk of samples within that node.[1] The full likelihood (cf. section 3.2.2 on page 37) of a tree with terminal nodes $\mathcal{J}$ is

$$\prod_{k \in \mathcal{J}} \prod_{i \in \mathcal{I}_k} (h_0(y_i)\theta_k)^{\delta_i} \exp(-H_0(y_i)).$$

To estimate the baseline cumulative hazard function $H_0(\cdot)$ and the parameter $\theta_k$ an iterative procedure needs to be applied, starting with $\theta_k = 1$, $\forall k = 1, \ldots, K$,

$$\hat{H}_0(t) = \sum_{\{i | t_i \leq t\}} \frac{d_i}{\sum_{k \in \mathcal{J}} \sum_{j \in \mathcal{I}_k} I(y_j \geq t_i) \hat{\theta}_k}, \tag{3.114}$$

$$\hat{\theta}_k = \frac{\sum_{i \in \mathcal{I}_k} \delta_i}{\sum_{i \in \mathcal{I}_k} \hat{H}_0(y_i)}, \tag{3.115}$$

where $\hat{H}_0(t)$ is Breslow's estimator (see section 3.2.3 on page 42) based on the estimate $\hat{\theta}_k$ of the previous iteration. In [188], estimates were approximated by the result after the first iteration is completed. Finally, the improvement in deviance is computed in analogy to the reduction in variance of regular regression trees:

$$\rho_j = \frac{1}{n} \left( \sum_{i \in \mathcal{I}_P} \operatorname{dev}(i, P) - \sum_{i \in \mathcal{I}_L} \operatorname{dev}(i, L) - \sum_{i \in \mathcal{I}_R} \operatorname{dev}(i, R) \right), \tag{3.116}$$

$$\operatorname{dev}(i, k) = 2 \left[ \delta_i \log \left( \frac{\delta_i}{\hat{H}_0(y_i)\hat{\theta}_k} \right) - (\delta_i - \hat{H}_0(y_i)\hat{\theta}_k) \right], \tag{3.117}$$

[1] $\theta_k$ plays the same role as the exponential of a linear model $\exp(\boldsymbol{x}^\top \boldsymbol{\beta})$ in the Cox model.

where $\text{dev}(i, k)$ is the deviance residual of the $i$-th sample in node $k$ and $0 \cdot \log 0 = 0$. The split with the largest improvement is ultimately selected.

Zhang [329] explored a different idea to define a split criterion for survival trees. Rather than basing the split criterion on well known identities in survival analysis such as the log-rank test or Cox model, their motivation comes from "the observation that an ideally homogeneous node should consist of subjects whose observed times are close and who are mostly censored or mostly uncensored" [329, p. 306]. They introduce an impurity measure that is a weighted sum of two elements: 1) a scaled variance estimator with respect to observed times, and 2) Shannon's entropy with respect to the event indicator:

$$\text{impurity}(k) = w_1 \frac{\sum_{i \in \mathcal{I}_k} (y_i - \mu_y(k))^2}{\sum_{i \in \mathcal{I}_k} y_i^2} + w_2 \left(-p_k \log_2 p_k - (1 - p_k) \log_2 (1 - p_k)\right), \tag{3.118}$$

where $w_1$, $w_2$ are hyper-parameters and $p_k = n_k^{-1} \sum_{i \in \mathcal{I}_k} (1 - \delta_i)$ is the proportion of censoring in node $k$. The split that results in the highest reduction in impurity is selected.

The split criterion proposed in [157, 159] is motivated from the conservation-of-events principle, which states "that the sum of the estimated cumulative hazard function over the observed time points (deaths and censored values) must equal the total number of deaths" [157, p. 26]. The deviation from the conservation-of-events principle in node $k$ is measured by

$$\Delta\text{CoE}(k, y_{\max}) = |\sum_{i \in \mathcal{I}_k} I(y_i \leq y_{\max}) \hat{H}_k(y_i) - \sum_{i \in \mathcal{I}_k} I(y_i \leq y_{\max}) \delta_i|. \tag{3.119}$$

By iterating over all possible values for $y_{\max}$ for each child node, their split criterion determines how well the conservation-of-events principle is preserved. Assuming samples in each node are sorted such that $y_1 \leq y_2 \leq \cdots \leq y_{n_k}$, then the split criterion can be computed by

$$\rho_j = \left(1 + \frac{1}{n_P} \sum_{k \in \{L, R\}} n_k \sum_{i=1}^{n_k - 1} \Delta\text{CoE}(k, y_i)\right)^{-1}. \tag{3.120}$$

The inverse is applied, because well separated child nodes would result in small values for $\Delta\text{CoE}(k, y_i)$. The split with the maximum value for $\rho_j$ is selected $(j = 1, \ldots, p)$.

Jin *et al.* [162] proposed a splitting criterion that finds a split such that the difference in restricted mean survival time $\text{RMST}(t) = \int_0^t S(u) du$ is maximized between child nodes. The change in within-node variance of restricted mean survival time can be computed by

$$\rho_j = \frac{n_L n_R}{n_P (n_L + n_R)} \left(\int_0^{t_L^{\max}} \hat{S}_L(u) du - \int_0^{t_R^{\max}} \hat{S}_R(u) du\right)^2, \tag{3.121}$$

where $\hat{S}_L$ is the Kaplan-Meier estimate (2.19) of the survival function in the left child node and $t_L^{\max} = \max\{y_i \mid i \in \mathcal{I}_L\}$ is the maximum observed time point in the left child node (analogous for $\hat{S}_R$ and $t_R^{\max}$ with respect to the right child node).

Schmid *et al.* [255] suggested to use Harrell's concordance index – a common evaluation measure for survival models (cf. section 3.7.1 on page 84) – as split criterion. To determine the goodness of a split with respect to the $j$-th feature, the ratio of the number of concordant to discordant pairs is computed with respect to the binary risk estimate $\hat{f}(\boldsymbol{x}_u) = I(u \in \mathcal{I}_R) = I(x_{uj} > \tau)$:

$$\begin{aligned} \hat{c}_j &= \frac{\sum_{u \neq v} \delta_u I(y_u < y_v) \Psi(u, v)}{\sum_{u \neq v} \delta_u I(y_u < y_v)}, \\ \Psi(u, v) &= \begin{cases} 1 & \text{if } x_{uj} > \tau \text{ and } x_{vj} \leq \tau, \\ 0.5 & \text{if } I(x_{uj} > \tau) = I(x_{vj} > \tau), \\ 0 & \text{else.} \end{cases} \end{aligned} \tag{3.122}$$

The function $\Psi(u, v)$ accounts for tied risk estimates, i.e., samples assigned to the same child node, by assigning them the value 0.5. The final split criterion is $\rho_j = \max(1 - \hat{c}_j, \hat{c}_j)$. In their experiments, Schmid *et al.* compared their proposed split criterion to log-rank splitting in a random survival forest (see section 3.6 on page 79). They concluded that splitting based on Harrell's concordance index is superior to log-rank splitting for datasets that are characterized by small sample size and high amount of censoring, whereas it was inferior when many unrelated features existed and the signal-to-noise ratio was small.

In [56], median regression trees for right censored survival data have been proposed. Their split criterion is based on the improvement in absolute distance to the within-node median survival time. To account for censoring, they impute survival times of censored patients using the Buckley-James estimator (see section 3.1.1 on page 33). Instead of employing greedy search to find the optimal feature and cut point of a split, they separate the two tasks following the approach of Loh and Shih [197].

Finally, Molinaro *et al.* [214] proposed a general framework to construct a survival tree based on inverse probability of censoring (IPC) weights (see section 3.1.2 on page 34). Therefore, building a survival tree is identical to building a regression tree, except that each sample is assigned an IPC weight. Nodes are split based on the within-node variance as impurity measure, which is the standard for regression trees [39].

### 3.5.3 Missing Data

Many real-world datasets are affected by missing values: the complete list of features is only available for a subset of samples in the dataset and for the remaining samples one or more features have not been recorded and therefore are missing. Commonly, missing values are addressed in a pre-processing step that fills in the missing values, but tree-based methods can be adapted to be aware of missing values during training and prediction.

Missing values in categorical variables can be accounted for by introducing an additional “missing” category, which eliminates all missing values and training can continue as usual. Alternatively, *surrogate splits* are applicable to both continuous-valued and categorical features [39, 130]. During training, the quality of a split based on the $j$-th feature is only evaluated with respect to samples where the $j$-th feature is available. Once the optimal feature with respect to the split criterion is found, one or more surrogate splits are defined that mimic that split. A surrogate split should approximate the split on the original feature and cut point, but use a different feature and cut point, thus exploiting correlations between features.

Let $j^*$ and $\tau^*$ denote the feature and cut point corresponding to the optimal split. Finding a surrogate split can be formulated as a binary classification problem, where samples are assigned labels according to which child node they were assigned to. Accordingly, the objective is to find an alternative feature $j$ and cut point $\tau$ that minimize the misclassification error [285]:

$$\operatorname*{argmin}_{\tau, j \neq j^*} \frac{1}{n_P} \left[ \sum_{i \in \mathcal{I}_L} I(x_{ij} > \tau) + \sum_{i \in \mathcal{I}_R} I(x_{ij} \leq \tau) \right], \tag{3.123}$$

where $\mathcal{I}_L$ and $\mathcal{I}_R$ are computed with respect to the optimal split based on feature $j^*$ and cut point $\tau^*$. In addition to computing the error of possible surrogate splits, a “blind split” is considered as fallback. A “blind split” assigns a sample always to the child node with the higher number of samples, and its misclassification error is $\min(n_L/n_P, n_R/n_P)$ [285]. Suitable surrogate splits are given by all features and cut points with misclassification error less than the “blind split”.

The list of surrogate splits for each node is stored such that it can be accessed during prediction. If a new sample with missing covariate $j$ reaches a node that splits based on $j$, the surrogate split with the smallest misclassification error (3.123) is applied, or if that feature is missing as well, the surrogate split with second smallest misclassification error, and so forth. If the features of all possible surrogate splits are missing, a “blind split” is carried out as the last resort.

## 3.6 Random Survival Forest

### 3.6.1 Ensemble of Survival Trees

A random forest is an ensemble of multiple, de-correlated tree-based learners; it was originally proposed for classification and regression tasks by Breiman [37]. Construction of individual trees in a random forest closely follows the process for regular classification and regression trees with some key differences to ensure that trees are de-correlated: 1) each tree is built on a different bootstrap sample of the original training data, and 2) at each node, the split criterion is only evaluated for a randomly selected subset of features and thresholds. Randomizing the computation of the split criterion is the main reason why construction a survival tree remains computationally feasible even with high-dimensional data. Predictions are formed by aggregating predictions of individual trees in the ensemble.

Ishwaran *et al.* [158] created an ensemble of relative risk trees [188] following Breiman's random forest framework [37]. Their ensemble consisted of many unpruned trees, which were constructed as described by LeBlanc and Crowley [188] (their split criterion using the deviance of a full likelihood Cox model was described in the previous section). The prediction $\hat{f}(\boldsymbol{x})$ of a single tree corresponds to the full relative risk estimate $\hat{\theta}_k$ associated with the leaf node reached by $\boldsymbol{x}$. A full relative risk estimate is obtained by repeatedly applying eqs. (3.114) and (3.115) until convergence. The ensemble prediction is the average relative risk prediction across all trees.

Hothorn *et al.* [148] constructed a random forest for survival analysis by using inverse probability of censoring weights when constructing individual trees [214]. Since leaf models correspond to mean log survival times of samples in that leaf, the ensemble prediction $\hat{f}(\boldsymbol{x})$ is the weighted average of mean log survival times with respect to all samples in leafs reached by $\boldsymbol{x}$.

Ishwaran *et al.* [159] proposed *random survival forests* using a split criterion and aggregation procedure inspired by the conservation-of-events principle described in section 3.5.2 (page 73). After training $B$ trees on $B$ bootstrap samples of the original training data, each terminal node of each tree contains an estimate $\hat{H}_k(t)$ of the cumulative hazard function via the Nelson-Aalen estimator (2.20). Let $\hat{H}_b(t|\boldsymbol{x})$ denote the estimated cumulative hazard function in the leaf node of the $b$-th tree that $\boldsymbol{x}$ was assigned to. The ensemble prediction $\hat{f}_{\mathrm{RSF}}(\boldsymbol{x})$ (referred to as *ensemble mortality* in [159]) can be obtained by averaging estimated cumulative hazard functions from all trees in the ensemble and computing the sum of the averaged CHF over all $m$ observed time points:

$$\hat{f}_{\mathrm{RSF}}(\boldsymbol{x}) = \sum_{i=1}^{m} \frac{1}{B} \sum_{b=1}^{B} \hat{H}_b(y_i|\boldsymbol{x}). \tag{3.124}$$

Due to the conservation-of-events principle, $\hat{f}_{\mathrm{RSF}}(\boldsymbol{x})$ is an estimate of the expected total number of deaths for subjects with covariates $\boldsymbol{x}$.

### 3.6.2 Variable Importance

In clinical research, the impact and the role of covariates in a disease are often of primary interest, whereas the mere predictive capability of a model plays a secondary role. Information about variable importance is readily available from linear models such as Cox's proportional hazards model (see section 3.2 on page 35), because each feature is assigned a weight that directly reflects its importance (assuming there are no scale differences between features). Although the random survival forest is likely to achieve higher predictive performance than a linear model, the influence of features in the decision process are not directly accessible. In particular, if data is to be analyzed that consists of a high number of features with little or no impact on survival, studying feature importances provides valuable information about the most important factors in a disease. In this section, I will describe basic measures of variable importance that can be retrieved from any random survival forest (or any other type of forest).

A crude measure of variable importance arises when simply counting the number of times a variable was selected for a split. A slightly more powerful measure can be obtained by considering the value of the split statistic $\rho_j$ at each split and tree. The importance of variable $j$ is the sum of split statistics $\rho_j$, aggregated over all nodes of all trees [132]. Breiman [37] suggested an alternative measure of variable importance based on out-of-bag (OOB) samples, i.e., samples that were not part of a particular bootstrap sample a tree in the ensemble was trained with. First, OOB samples are dropped down the respective trees of the forest to obtain an estimate $E_{\text{OOB}}$ of the ensemble error. Next, the values of the $j$-th feature are randomly permuted to remove any correlation to the outcome variable and the forest is applied to the altered OOB samples to obtain the error $E_j$. If variable $j$ is a good predictor of the outcome, the error $E_j$ is going to be significantly larger than $E_{\text{OOB}}$. Consequently, the variable importance of the $j$-th feature is the difference between the OOB error before and after permutation, $E_{\text{OOB}} - E_j$.

Note that the forest is not re-trained after permuting variable $j$; any connection between the $j$-th feature and the outcome is only broken in the OOB samples, not during construction of the forest. An alternative description of the permutation procedure that emphasizes this is that an OOB sample is randomly assigned to a child node if the split depends on the $j$-th feature [159]. If one would re-train the forest, $E_{\text{OOB}} - E_j$ would be the difference in prediction error between a forest that had access to the $j$-th feature and a forest where the $j$-th feature was unavailable. If the dataset contains a variable that is correlated with the $j$-th feature, re-training would probably compensate for the removal of the $j$-th feature by increasing the importance of the other variable, and $E_{\text{OOB}} - E_j$ would be small [159].

In all cases, the importance of a variable depends on the choice of split criterion. In particular, if a split criterion is biased towards variables of certain properties, the measure of variable importance will be biased as well. For instance, it is well documented that using the Gini index as split criterion for classification yields a biased variable importance measure in the presence of variables with many categories or with high correlation to other features [221, 222, 279, 280].

### 3.6.3 Missing Data

A random forest can be applied to samples with missing values by computing surrogate splits (see section 3.5.3 on page 78) in the individual trees of the ensemble. However, using surrogate splits in a forest can be problematic because of the randomness in selecting candidate variables in each node and the computational costs involved in finding surrogate splits. Instead, Breiman and Cutler [38] proposed an imputation approach that relies on the proximity measure defined in eq. (3.125) below.

> **Definition 3.10: Proximity.** Let $B$ be the number of trees and $\mathcal{I}_{b,v}$ the set of samples that arrived at the $v$-th leaf node of the $b$-th tree. The *proximity* between samples $i$ and $j$ is the number of times feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ coincide in the same leaf node:
>
> $$\operatorname{prox}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{B} \sum_{b=1}^{B} \sum_{v=1}^{J_b} I(\boldsymbol{x}_i \in \mathcal{I}_{b,v} \land \boldsymbol{x}_j \in \mathcal{I}_{b,v}), \tag{3.125}$$
>
> where $J_b$ is the total number of leaf nodes of the $b$-th tree.

Missing values are imputed by iteratively constructing a random forest based on imputed data, starting with a rough imputation. For instance, missing values in continuous features can be imputed by the mean or median of all non-missing samples, and categorical variables can be imputed by the mode of all non-missing samples. After training a random forest with the imputed values, the missing values are re-imputed based on the proximity (3.125) between samples in the data. Continuous values are imputed by the proximity-weighted average of all non-missing samples, and categorical values are imputed by the value that occurs most frequently among all non-missing samples, where frequencies are proximity weighted. Now, the forest is re-trained again and the imputation procedure is repeated until imputed values converge to a stable estimate.

Ishwaran *et al.* [159] argued that the imputation method by Breiman and Cutler [38] does not account for missing values during testing and that it results in biased OOB error and variable importance estimates. The method by Ishwaran *et al.* [159] imputes missing values from non-missing in-bag samples each time a node is split. For simplicity, I assume that only a single feature contains missing values, but the algorithm below can be easily extended if more than one feature has missing values.

For the $k$-th node in a tree with $K$ nodes, let $\boldsymbol{X}^{(k,j)}$ denote a vector of non-missing values of the $j$-th feature from in-bag samples reaching node $k$. When growing a tree, at each node, the missing values in the $j$-th feature are imputed by randomly drawing a value from the empirical distribution function of $\boldsymbol{X}^{(k,j)}$. After the optimal split has been determined, missing values are reset in the child nodes and imputation is repeated as above. During testing, a new sample $\boldsymbol{x}$ is sent down each tree in the ensemble until a split that depends on a missing feature in $\boldsymbol{x}$ is encountered. If the $j$-th feature is missing, its value is imputed by randomly drawing from the same empirical distribution function of $\boldsymbol{X}^{(k,j)}$ used during training. The final imputation consists of the average imputed value across all trees (for continuous features), or the most frequently imputed value (for categorical features). Depending on the amount of missing values, the imputation process by Ishwaran *et al.* [159] can be iteratively repeated as in [38] to improve imputation accuracy.

## 3.7 Evaluation Measures

Once a survival model has been trained, it is necessary to determine how well the model is expected to perform on new data that was not used for building the model.

> **Definition 3.11: True error rate and expected true error.** Let $\hat{f}_{\mathcal{D}}$ denote a model that was estimated from data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^n$, and $L$ a loss function that measures the discrepancy between the actual outcome $y_0$ and the predicted outcome $\hat{f}_{\mathcal{D}}(\boldsymbol{x}_0)$. Following [131], the *true error rate* of $\hat{f}_{\mathcal{D}}$ on a new example $(\boldsymbol{x}_0, y_0)$ drawn from the joint distribution of $(X, Y)$ is
>
> $$\mathrm{Err}(\hat{f}_{\mathcal{D}}) = E_{(\boldsymbol{x}_0, y_0) \sim (X,Y)} \left[ L(y_0, \hat{f}_{\mathcal{D}}(\boldsymbol{x}_0)) \mid \hat{f}_{\mathcal{D}}, \mathcal{D} \right], \qquad (3.126)$$
>
> where the model $\hat{f}_{\mathcal{D}}$ and the training data $\mathcal{D}$ are fixed and only $(\boldsymbol{x}_0, y_0)$ is random. Alternative names for the true error rate are *generalization error*, *prediction error*, *test error* or *extra-sample error* [82, 84, 131]. Averaging the true error rate over training sets $\mathcal{D}$, yields the *expected true error*
>
> $$\overline{\mathrm{Err}}(\hat{f}_{\mathcal{D}}) = E_{\mathcal{D}}(\mathrm{Err}(\hat{f}_{\mathcal{D}})). \qquad (3.127)$$

Estimating (3.126) would be preferred, because it provides information on how well a model trained on samples in $\mathcal{D}$ generalizes to an independent sample $\boldsymbol{x}_0$. However, methods such as cross-validation and the bootstrap estimate the expected true error (3.127), i.e., they consider the expectation over multiple training sets rather than a fixed training set [82, 84, 131]. In this section, I will focus on characteristics of $L$ to assess the difference between actual survival times and predicted survival times. Due to censoring – it is only possible to observe $Y = \min(T, C)$ – regular metrics such as the root mean squared error cannot be employed. Hence, evaluation measures for survival models need to be aware of censoring.

### 3.7.1 Concordance Index

The concordance index ($c$ index) is a measure of rank correlation between predictions $\hat{f}(\boldsymbol{x})$ and observed time points $y$ that is closely related to Kendall's $\tau$ [127, 128]. The concordance index by Harrell *et al.* [127, 128] is the ratio of correctly ordered (concordant) pairs to comparable pairs. Two patients $i$ and $j$ are *comparable* if the patient with lower observed time $y$ experienced an event, i.e., if $y_i < y_j$ and $\delta_i = 1$. The set $\mathcal{P} = \{(i, j) \mid y_i < y_j \wedge \delta_i = 1\}_{i,j=1}^n$ comprises all comparable pairs. A comparable pair $(i, j)$ is *concordant* if the estimated risk by a survival model $\hat{f}$ is higher for subjects with lower survival time, i.e., $\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j) \wedge y_i < y_j$, otherwise the pair is *discordant*. Harrell *et al.* [127, 128] proposed to estimate the probability $P(\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j) \mid t_i < t_j)$ by the ratio of concordant to comparable pairs by

$$\hat{c}_{\text{Harrell}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} I(\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j)). \tag{3.128}$$

The $c$ index is within the interval $[0; 1]$ and is identical to the area under the receiver operating characteristics (ROC) curve [125, 232] if the outcome is binary and no censoring is present [127]. A random model has a $c$ index of 0.5 and an optimal model of 1.0.

Uno *et al.* [292] observed that the estimate $\hat{c}_{\text{Harrell}}$ depends on the distribution of censoring times in the test data, which leads to biased estimates of the true concordance index. They addressed this problem by including inverse probability of censoring weights (see section 3.1.2 on page 34) into the estimator. Instead of estimating the probability $P(\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j) \mid t_i < t_j)$, they estimated the truncated probability $P(\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j) \mid t_i < t_j \wedge t_i < \tau)$, where $\tau > 0$ corresponds to a manually defined time point such that the probability of censoring is guaranteed to be non-zero: $P(C > \tau) > 0$. The set of comparable samples in the interval $[0; \tau]$ is defined as $\mathcal{P}_\tau = \{(i, j) \mid y_i < y_j \wedge y_i < \tau \wedge \delta_i = 1\}_{i,j=1}^n$. They constructed their estimator by weighting each sample in the comparison by the inverse probability of censoring weight $\omega_i$ as defined in eq. (3.18):

$$\hat{c}_{\text{Uno}}(\tau) = \frac{1}{\sum\limits_{(i,j) \in \mathcal{P}_\tau} \omega_i^2} \sum_{(i,j) \in \mathcal{P}_\tau} \omega_i^2 I(\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j)). \tag{3.129}$$

To compute the weights $\omega_i$, they assumed that censoring is independent of the features (random censoring) and used the non-parametric Kaplan-Meier estimator (2.19) to estimate the conditional censoring survivor function. Uno *et al.* [292] showed that $\hat{c}_{\text{Uno}}(\tau)$ is a consistent estimator.

The estimators (3.128) and (3.129) do not account for situations when risk scores are tied, but both can be easily extended to account for ties by replacing the indicator function in the numerator with

$$\Psi(i, j) = \begin{cases} 1 & \text{if } \hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j), \\ 0.5 & \text{if } \hat{f}(\boldsymbol{x}_i) = \hat{f}(\boldsymbol{x}_j) \text{ and } i \neq j, \\ 0 & \text{else.} \end{cases} \tag{3.130}$$

### 3.7.2 Time-dependent Area under the Curve

The area under the receiver operating characteristics curve (ROC) is a popular performance measure for binary classification tasks [125, 232]. In the medical domain, it is often used to determine how well estimated risk scores can separate diseased patients (cases) from healthy patients (controls). Given a model $\hat{f}$ that predicts continues risk scores, i.e., $\hat{f}(\boldsymbol{x}) \in \mathbb{R}$, the ROC curve plots the false positive rate (1 - specificity) against the true positive rate (sensitivity) for every threshold $\tau \in \mathbb{R}$:

$$\begin{aligned} \mathrm{Se}(\tau) &= P(\hat{f}(\boldsymbol{x}_i) > \tau \mid d_i = 1), \\ \mathrm{Sp}(\tau) &= P(\hat{f}(\boldsymbol{x}_i) \leq \tau \mid d_i = 0), \\ \mathrm{ROC} &= \{(1 - \mathrm{Sp}(\tau), \mathrm{Se}(\tau))\}_{\tau \in \mathbb{R}}, \end{aligned}$$

where $\mathrm{Se}(\tau)$ and $\mathrm{Sp}(\tau)$ represent the sensitivity and specificity at threshold $\tau$, and $d_i = 1$ and $d_i = 0$ indicate that the $i$-th patient is diseased and healthy. Sensitivity and specificity can be estimated from a confusion table as

$$\begin{aligned} \widehat{\mathrm{Se}}(\tau) &= \frac{\sum_{i=1}^{n} I(\hat{f}(\boldsymbol{x}_i) > \tau) I(d_i = 1)}{\sum_{i=1}^{n} I(d_i = 1)} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}, \\ \widehat{\mathrm{Sp}}(\tau) &= \frac{\sum_{i=1}^{n} I(\hat{f}(\boldsymbol{x}_i) \leq \tau) I(d_i = 0)}{\sum_{i=1}^{n} I(d_i = 0)} = \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}}. \end{aligned}$$

When extending the ROC curve to continuous outcomes, in particular survival time, a patient's disease status is typically not fixed and changes over time: at enrollment a subject is usually healthy, but may be diseased at some later time point. Consequently, sensitivity and specificity become time-dependent measures. Several definitions of time-dependent sensitivity and specificity exist in the literature; they differ in the definition of "cases" and "controls" at time point $t$.

Following Heagerty *et al.* [134] and Heagerty and Zheng [135], *incident cases* are defined as individuals that experience an event at time $t$ ($t_i = t$) and *cumulative cases* as individuals that experienced an event prior to or at time $t$ ($t_i \leq t$). In addition, *static controls* are all subjects that experienced an event after a fixed time point $t^*$ ($t_i > t^*$) and *dynamic controls* are those with $t_i > t$. Based on the aforementioned definitions, three different time-dependent ROC curves can be defined together with their corresponding area under the time-dependent ROC curve (AUROC).

Quantities based on incident and cumulative cases are indicated by a superscript $\mathbb{I}$ and $\mathbb{C}$, respectively, and quantities based on static and dynamic controls by a superscript $\mathbb{S}$ and $\mathbb{D}$.

**Definition 3.12: Cumulative-dynamic ROC curve [134].**

$$\begin{aligned}\mathrm{Se}^{\mathbb{C}}(\tau, t) &= P(\hat{f}(\boldsymbol{x}_i) > \tau \mid t_i \leq t),\\ \mathrm{Sp}^{\mathbb{D}}(\tau, t) &= P(\hat{f}(\boldsymbol{x}_i) \leq \tau \mid t_i > t),\\ \mathrm{ROC}^{\mathbb{C}/\mathbb{D}}(t) &= \{(1 - \mathrm{Sp}^{\mathbb{D}}(\tau), \mathrm{Se}^{\mathbb{C}}(\tau))\}_{\tau \in \mathbb{R}},\\ \mathrm{AUROC}^{\mathbb{C}/\mathbb{D}}(t) &= P(\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j) \mid t_i \leq t \wedge t_j > t).\end{aligned} \tag{3.131}$$

**Definition 3.13: Incident-dynamic ROC curve [135].**

$$\begin{aligned}\mathrm{Se}^{\mathbb{I}}(\tau, t) &= P(\hat{f}(\boldsymbol{x}_i) > \tau \mid t_i = t),\\ \mathrm{ROC}^{\mathbb{I}/\mathbb{D}}(t) &= \{(1 - \mathrm{Sp}^{\mathbb{D}}(\tau), \mathrm{Se}^{\mathbb{I}}(\tau))\}_{\tau \in \mathbb{R}},\\ \mathrm{AUROC}^{\mathbb{I}/\mathbb{D}}(t) &= P(\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j) \mid t_i = t \wedge t_j > t).\end{aligned} \tag{3.132}$$

**Definition 3.14: Incident-static ROC curve [89, 268].**

$$\begin{aligned}\mathrm{Sp}^{\mathbb{S}}(\tau, t) &= P(\hat{f}(\boldsymbol{x}_i) \leq \tau \mid t_i > t^*),\\ \mathrm{ROC}^{\mathbb{I}/\mathbb{S}}(t) &= \{(1 - \mathrm{Sp}^{\mathbb{S}}(\tau), \mathrm{Se}^{\mathbb{I}}(\tau))\}_{\tau \in \mathbb{R}},\\ \mathrm{AUROC}^{\mathbb{I}/\mathbb{S}}(t) &= P(\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j) \mid t_i = t \wedge t_j > t^*).\end{aligned} \tag{3.133}$$

I will now focus on the *cumulative-dynamic* ROC curve and briefly present the estimator of $\mathrm{AUROC}^{\mathbb{C}/\mathbb{D}}(t)$ proposed in [153, 293]. The estimators of cumulative-dynamic sensitivity and specificity by Hung and Chiang [153] and Uno *et al.* [293] are based on assigning observations inverse probability of censoring (IPC) weights (see section 3.1.2 on page 34), which leads to

$$\widehat{\mathrm{Se}}^{\mathbb{C}}(\tau, t) = \frac{\sum_{i=1}^n I(\hat{f}(\boldsymbol{x}_i) > \tau) I(y_i \leq t) \omega_i}{\sum_{i=1}^n I(y_i \leq t) \omega_i}, \tag{3.134}$$

$$\widehat{\mathrm{Sp}}^{\mathbb{D}}(\tau, t) = \frac{\sum_{i=1}^n I(\hat{f}(\boldsymbol{x}_i) \leq \tau) I(y_i > t)}{\sum_{i=1}^n I(y_i > t)}. \tag{3.135}$$

The estimator of the time-dependent specificity simplifies to the naïve estimator because IPC weights correspond to $1/\hat{G}(t)$ for all observations and cancel each other out [29]. Using the result of Satten and Datta [250], the denominator in the estimator of the sensitivity (3.134) is equivalent to $n(1 - \hat{S}_T(t))$, because $n^{-1} \sum_{i=1}^n I(y_i \leq t)\omega_i = 1 - \hat{S}_T(t)$, with $\hat{S}_T(t)$ being the Kaplan-Meier estimator (2.19) of $P(T > t)$. Moreover, $\hat{S}_Y(t) = n^{-1} \sum_{i=1}^n I(y_i > t)$ denotes the estimator of $P(Y > t)$. By substituting both

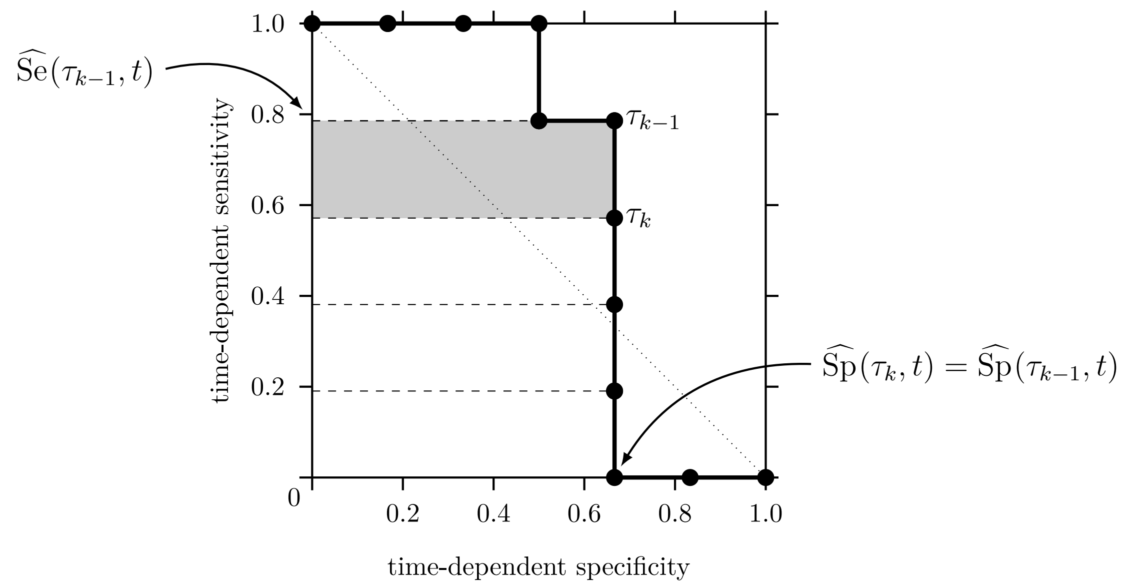


**Figure 3.2**: Example of time-dependent ROC curve. Bullets indicate individual points on the ROC curve with the associated threshold $\tau$. Dashed horizontal lines mark the contribution of false negatives to the area under the ROC curve. The dotted line represents a random model.

formulations into eqs. (3.134) and (3.135), the estimators become

$$\widehat{\mathrm{Se}}^{\mathbb{C}}(\tau, t) = \frac{\sum_{i=1}^{n} I(\hat{f}(\boldsymbol{x}_i) > \tau) I(y_i \leq t) \omega_i}{n(1 - \hat{S}_T(t))}, \tag{3.136}$$

$$\widehat{\mathrm{Sp}}^{\mathbb{D}}(\tau, t) = \frac{\sum_{i=1}^{n} I(\hat{f}(\boldsymbol{x}_i) \leq \tau) I(y_i > t)}{n\hat{S}_Y(t)}. \tag{3.137}$$

The area under the cumulative-dynamic ROC curve $\mathrm{AUROC}^{\mathbb{C}/\mathbb{D}}(t)$ can be estimated from $\widehat{\mathrm{Se}}^{\mathbb{C}}(\tau, t)$ and $\widehat{\mathrm{Sp}}^{\mathbb{D}}(\tau, t)$ by computing the sum over rectangular areas with width corresponding to the increase in specificity and height corresponding to the loss in sensitivity (see fig. 3.2). If there are no ties in observed time points $y_i$ and in predicted scores $\hat{f}(\boldsymbol{x}_i)$ for all $i = 1, \ldots, n$, the area under the curve increases with every false negative and remains the same for each false positive. Thus, a single increment of the area under the ROC curve is equivalent to the area of a rectangle with width $\widehat{\mathrm{Sp}}^{\mathbb{D}}(\tau_k, t)$ and height $\widehat{\mathrm{Se}}^{\mathbb{C}}(\tau_k, t) - \widehat{\mathrm{Se}}^{\mathbb{C}}(\tau_{k-1}, t)$ (see gray box in fig. 3.2). Assuming $\tau_k > \tau_{k-1}$, the

$k$-th update of the area under the ROC curve has the form

$$\begin{aligned}
&\left(\widehat{\mathrm{Se}}^{\mathbb{C}}(\tau_{k-1}, t) - \widehat{\mathrm{Se}}^{\mathbb{C}}(\tau_k, t)\right) \widehat{\mathrm{Sp}}^{\mathbb{D}}(\tau_k, t) \\
&= \left[ \frac{\sum_{i=1}^n I(\hat{f}(\boldsymbol{x}_i) > \tau_{k-1}) I(y_i \leq t)\omega_i - \sum_{i=1}^n I(\hat{f}(\boldsymbol{x}_i) > \tau_k) I(y_i \leq t)\omega_i}{n(1 - \hat{S}_T(t))} \right] \widehat{\mathrm{Sp}}^{\mathbb{D}}(\tau_k, t) \\
&= \left[ \frac{\sum_{i=1}^n \left( I(\hat{f}(\boldsymbol{x}_i) > \tau_{k-1}) - I(\hat{f}(\boldsymbol{x}_i) > \tau_k) \right) I(y_i \leq t)\omega_i}{n(1 - \hat{S}_T(t))} \right] \widehat{\mathrm{Sp}}^{\mathbb{D}}(\tau_k, t) \\
&= \left( \frac{I(y_k \leq t)\omega_k}{n(1 - \hat{S}_T(t))} \right) \widehat{\mathrm{Sp}}^{\mathbb{D}}(\tau_k, t) \\
&= \left( \frac{I(y_k \leq t)\omega_k}{n(1 - \hat{S}_T(t))} \right) \left( \frac{\sum_{i=1}^n I(\hat{f}(\boldsymbol{x}_i) \leq \tau_k) I(y_i > t)}{n\hat{S}_Y(t)} \right),
\end{aligned}$$

where $I(\hat{f}(\boldsymbol{x}_i) > \tau_k) - I(\hat{f}(\boldsymbol{x}_i) > \tau_{k-1})$ is one for $i = k$ and zero otherwise. The latter arises from the assumption that risk scores are unique: when going from $\tau_{k-1} = \hat{f}(\boldsymbol{x}_{k-1})$ to the next greater threshold $\tau_k = \hat{f}(\boldsymbol{x}_k)$, only the $k$-th sample switches from being predicted negative to being predicted positive and all other samples remain unchanged.

Finally, an estimator of the area under the cumulative-dynamic ROC curve can be obtained by iterating over all $n$ unique thresholds and summing up the corresponding rectangular areas [153, 293]:

$$\begin{aligned}
\widehat{\mathrm{AUROC}}^{\mathbb{C}/\mathbb{D}}(t) &= \sum_{i=1}^n \left( \widehat{\mathrm{Se}}(\hat{f}(\boldsymbol{x}_{i-1}), t) - \widehat{\mathrm{Se}}(\hat{f}(\boldsymbol{x}_i), t) \right) \widehat{\mathrm{Sp}}(\hat{f}(\boldsymbol{x}_i), t) \\
&= \sum_{i=1}^n \left( \frac{I(y_i \leq t)\omega_i}{n(1 - \hat{S}_T(t))} \right) \left( \frac{\sum_{j=1}^n I(\hat{f}(\boldsymbol{x}_j) \leq \hat{f}(\boldsymbol{x}_i)) I(y_j > t)}{n\hat{S}_Y(t)} \right) \\
&= \frac{\sum_{i=1}^n \sum_{j=1}^n I(y_i \leq t) I(y_j > t) I(\hat{f}(\boldsymbol{x}_j) \leq \hat{f}(\boldsymbol{x}_i))\omega_i}{n^2 \hat{S}_Y(t)(1 - \hat{S}_T(t))}.
\end{aligned} \tag{3.138}$$

The area under the cumulative-dynamic ROC curve can also be summarized over multiple time points by integrating over all time points $t$ [183, 254]:

$$c^{\mathbb{C}/\mathbb{D}} = E_T(\mathrm{AUROC}^{\mathbb{C}/\mathbb{D}}(T)) = \int_t \mathrm{AUROC}^{\mathbb{C}/\mathbb{D}}(t) f(t) dt, \tag{3.139}$$

where $f(t)$ denotes the probability density function of survival time $T$.

*Note.* The above summary statistic is motivated by Heagerty and Zheng [135], who suggested the following summary statistic for the area under the *incident-dynamic* ROC curve:

$$c^{\mathbb{I}/\mathbb{D}} = \int_t \mathrm{AUROC}^{\mathbb{I}/\mathbb{D}} 2 f(t) S(t) dt. \tag{3.140}$$

They showed that $c^{\mathbb{I}/\mathbb{D}}$ equals the probability $P(\hat{f}(\boldsymbol{x}_i) > \hat{f}(\boldsymbol{x}_j) \mid t_i < t_j)$, which is estimated by the concordance index of Harrell *et al.* [127, 128] and Uno *et al.* [292] as well. Therefore, $c^{\mathbb{I}/\mathbb{D}}$ corresponds to the time-independent concordance index and provides an alternative interpretation of the concordance index.

### 3.7.3 Brier Score

Graf *et al.* [117] proposed the time-dependent Brier score for right censored data. Let $\hat{\pi}(t|\boldsymbol{x})$ be the predicted probability of remaining event-free up to time point $t$ for a patient with covariates $\boldsymbol{x}$. The time-dependent Brier score is the mean squared error at time point $t$:

$$E\left[(I(T > t) - \hat{\pi}(t|\boldsymbol{x}))^2\right].$$

Due to right censoring, they split the mean squared error into three terms:

1. $y_i \leq t \wedge \delta_i = 1$,
2. $y_i > t$,
3. $y_i \leq t \wedge \delta_i = 0$

In the first case, an event occurred before the time point $t$ and the indicator function $I(y_i > t)$ is zero; the squared error becomes $(0 - \hat{\pi}(t|\boldsymbol{x}))^2$. In the second case, an event or censoring occurs after time $t$, resulting in the squared error $(1 - \hat{\pi}(t|\boldsymbol{x}))^2$. In the last case, the sample was censored before time point $t$, which means it is unknown if an event occurred; an error cannot be calculated. Finally, to account for censoring, Graf *et al.* weighted individual contributions to the overall error by the inverse probability of censoring weight $1/\hat{G}(t)$, estimated by the Kaplan-Meier estimator (2.19):

$$\mathrm{BS}_{\mathrm{censored}}(t) = \frac{1}{n}\sum_{i=1}^{n} I(y_i \leq t \wedge \delta_i = 1)\frac{(0 - \hat{\pi}(t|\boldsymbol{x}_i))^2}{\hat{G}(y_i)} + I(y_i > t)\frac{(1 - \hat{\pi}(t|\boldsymbol{x}_i))^2}{\hat{G}(t)}, \tag{3.141}$$

where they assume no tied event times and random censoring. Note that $t$ has to be chosen such that $P(C > t) > 0$ is guaranteed, otherwise the result is undefined due to division by zero. To compute the time-dependent Brier score over multiple time points in the interval $[0; \tau]$, $\mathrm{BS}_{\mathrm{censored}}(t)$ can be integrated over some time-dependent weight function $w(t)$:

$$\mathrm{IBS} = \int_0^{\tau} \mathrm{BS}_{\mathrm{censored}}(t) dw(t). \tag{3.142}$$

Graf *et al.* [117] suggested two weight functions: $w(t) = t/\tau$, and $w(t) = (1 - \hat{S}(t))(1 - \hat{S}(\tau))^{-1}$, where $\hat{S}(t)$ is an estimate of the marginal survival function. For practical purposes, obtaining the estimate $\hat{w}(t)$ is straightforward if IBS is estimated by the trapezoidal rule [141].

# 4 Missing Data

Most learning algorithms assume that every sample has a valid value for every feature in the dataset. However, in clinical data this is often not the case: the amount of information varies between patients. A value can be missing for a myriad of reasons: a patient refused to answer a particular question or did not remember the exact answer to a question, a particular diagnostic test was not performed purposely or could not be carried out, recording or storing the information was erroneous, and so forth. In particular, if data is collected over a long period of time or from multiple institutions, missing values are common. In this scenario, missing data often occurs in blocks, for instance, if a diagnostic test was not available for patients that enrolled early in the study or one institution did not perform a particular set of measurements. To address the missing data problem, one can choose from three options:

1. discard all samples that contain one or more missing value (complete case analysis),
2. adopt learning algorithm to explicitly allow missing values in the data,
3. fill-in missing values (imputation).

Complete case analysis is the simplest approach, but also associated with several limitations that often render it inappropriate [195]. First of all, it reduces the number of samples that are available for training, which leads to higher uncertainty in a model's parameter estimates. Second, the model is going to be biased if dropped samples differ systematically from samples that remained in the training data. In choosing the second option, an existing method is adopted to be applicable to data with missing values. Therefore, it requires detailed knowledge about the method and leads to a solution that is specifically tailored to that particular survival model. Examples of this approach are surrogate splits in survival trees (see section 3.5.3 on page 78) and built-in imputation in random survival forests (see section 3.6.3 on page 82). In this chapter, I will focus on approaches addressing the missing data problem through imputation, in particular multiple imputation.

*Note.* Processes related to censoring and those related to missing data can be unified, which leads to the more general concept of *coarsened data* [137]. For the remainder of this chapter, I assume that only features, but not the outcome (survival time), contain missing values.

## 4.1 Missing Data Generating Processes

Rubin [246] described three general mechanisms that lead to missing data from a probabilistic point of view. He assumes a probability distribution $\boldsymbol{R}$ that generates missing values and formalizes the dependency structure between the missing value generating process, the observed data and the missing data.

Let $\boldsymbol{R}$ be a set of random variables that determine the mechanism of missingness. $\boldsymbol{R}$ can be treated as a matrix of size identical to the data $\boldsymbol{X}$ with elements either 1 or 0, depending on whether the corresponding value is observed or missing. A full dataset $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ can be partitioned into an observed part $\boldsymbol{X}_{\text{obs}}$ and a missing part $\boldsymbol{X}_{\text{mis}}$ via the indicator matrix $\boldsymbol{R}$. Moreover, let $\boldsymbol{Z}$ denote a set of features that have no missing values and $\boldsymbol{Y}$ the dependent variable(s).

> **Definition 4.1: Missing completely at random.** The missing data generating process is called *missing completely at random* (MCAR) if the pattern of missingness is independent of the observed data as well as the missing data [246]:
>
> $$P(\boldsymbol{R} \mid \boldsymbol{X}) = P(\boldsymbol{R}). \tag{4.1}$$

> **Definition 4.2: Missing at random.** If the pattern of missingness is only independent of the missing data, but not the observed data, the missing data generating process is called *missing at random* (MAR) [246]:
>
> $$P(\boldsymbol{R} \mid \boldsymbol{X}) = P(\boldsymbol{R} \mid \boldsymbol{X}_{\text{obs}}). \tag{4.2}$$

> **Definition 4.3: Missing not at random.** If neither MCAR nor MAR holds, the pattern of missingness depends on the observed data as well as the missing data and is called *missing not at random* (MNAR) [246]:
>
> $$P(\boldsymbol{R} \mid \boldsymbol{X}) = P(\boldsymbol{R} \mid \boldsymbol{X}_{\text{obs}}, \boldsymbol{X}_{\text{mis}}). \tag{4.3}$$

The missing completely at random (MCAR) assumption is the strongest, because missing values should be unrelated to any factors – observed or unobserved – in the data generation process. Moreover, MCAR is the only scenario where complete-case analysis does not lead to biased estimates [195]. In contrast, missing values in the missing not at random (MNAR) scenario depend on observed and unobserved variables and will lead to biased estimates if not accounted for [195, 246]. In the missing at random (MAR) mechanism, a missing value can only arise if it depends on another observed value, but not on the missing value itself. In particular, the set of observed data $\boldsymbol{X}_{\text{obs}}$ may be different for each patient and therefore missing values may depend on different factors. In addition, the missing data mechanisms above can be grouped into two groups: ignorable and non-ignorable missing data. Data that is MCAR or MAR is referred to as *ignorable* missing data, and data that is MNAR as *non-ignorable* or *informative* missing data.

**Example 4.1.** To provide an intuition about MCAR, MAR and MNAR, consider an example presented in [136] about a study investigating the efficacy of programs to educate people about cardiovascular risks at their work place [114]. The body mass index (BMI) of all participants in the study was measured at baseline, and 3, 6, and 12 months after the initial assessment. If the BMI could not be recorded because participants were attending an off-site meeting at the time the measurement was scheduled, the missing data would follow the MCAR mechanism. If instead some subjects with high BMI at baseline did not attend follow-up measurements, disregarding whether they put on weight or lost weight since the last measurement, the missing data would follow the MAR mechanism – the reason for missing a measurement was recorded. Finally, missing values are MNAR if participants decided not to attend a follow-up if they gained weight since the last measurement. In such a case, a missing value would arise due to unknown factors that have not been recorded.

In practice, the MCAR assumption is often unlikely to hold and most work focuses on the situation were missing data is due to the MAR mechanism. Unfortunately, given only observed data, the MAR and MNAR scenarios are indistinguishable from each other without additional assumptions, which means the MAR assumption is untestable [109, 195].

## 4.2 Multiple Imputation

Multiple imputation [246, 247] describes a class of algorithms that propose $m > 1$ plausible values for each missing value, resulting in $m$ datasets without missing values, which can subsequently be analyzed by traditional methods for complete data. In the end, estimates from individual models are combined to form an overall model. The main advantage of multiple imputation is that it accounts for the uncertainty about the imputed value.

Multiple imputation builds on the MAR assumption by drawing values from the conditional distribution $P(\boldsymbol{X}_{\text{mis}} \mid \boldsymbol{X}_{\text{obs}}, \boldsymbol{Z}, \boldsymbol{Y}, \hat{\boldsymbol{\theta}})$ to impute missing values, where $\hat{\boldsymbol{\theta}}$ are parameter estimates of interest. If only a single feature is affected by missing values, Rubin [246] described an approach that first constructs a maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ from all complete samples $\boldsymbol{X}_{\text{obs}}$ and then draws $m$ plausible parameters $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(m)}$ from the observed-data posterior distribution $P(\boldsymbol{\theta} \mid \boldsymbol{X}_{\text{obs}}, \boldsymbol{Z}, \boldsymbol{Y})$. Imputations for missing data are obtained by drawing from $P(\boldsymbol{X}_{\text{mis}} \mid \boldsymbol{X}_{\text{obs}}, \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}^{(k)})$ for $k = 1, \ldots, m$ [252]. For instance, if the vector $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\beta}}_{\text{MLE}} \in \mathbb{R}^p$ contains estimated coefficients of a regression model and $\boldsymbol{U} \in \mathbb{R}^{p \times p}$ is the corresponding covariance matrix, the posterior distribution $P(\boldsymbol{\beta} \mid \boldsymbol{X}_{\text{obs}}, \boldsymbol{Z}, \boldsymbol{Y})$ can be approximated by the multivariate normal distribution: $\boldsymbol{\theta}^{(k)} \sim \mathcal{N}_p(\hat{\boldsymbol{\beta}}_{\text{MLE}}, \boldsymbol{U})$ [246]. In more complex settings, techniques such as Markov Chain Monte Carlo have to be used to draw from $P(\boldsymbol{\beta} \mid \boldsymbol{X}_{\text{obs}}, \boldsymbol{Z}, \boldsymbol{Y})$.

Finally, the process above is repeated $m$ times, yielding $m$ datasets that differ in the imputed values.

After generating $m$ datasets without missing values, traditional survival models for complete data can be trained on each of the $m$ datasets. Since each model will have its own estimated parameters, the final task consists of combining those estimates into an overall estimate. Rubin's rule [246] defines a procedure to pool the estimates $\hat{\boldsymbol{\theta}}^{(k)}$ and their covariance matrices $\boldsymbol{U}^{(k)}$ from $m$ imputed datasets under the assumption that

$$(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{U}),$$

where $\boldsymbol{\theta}$ are the unknown population parameters, $\hat{\boldsymbol{\theta}}$ their complete data estimates and $\boldsymbol{U}$ the associated variance estimate. The overall estimate $\bar{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ from $m$ imputed datasets is

$$\bar{\boldsymbol{\theta}} = \frac{1}{m} \sum_{k=1}^{m} \hat{\boldsymbol{\theta}}^{(k)}, \tag{4.4}$$

and the associated covariance matrix can be obtained by calculating the within-imputation variance $\bar{\boldsymbol{U}}$ and the between-imputation variance $\boldsymbol{B}$:

$$\bar{\boldsymbol{U}} = \frac{1}{m} \sum_{k=1}^{m} \boldsymbol{U}^{(j)} \tag{4.5}$$

$$\boldsymbol{B} = \frac{1}{m-1} \sum_{k=1}^{m} (\hat{\boldsymbol{\theta}}^{(k)} - \bar{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}}^{(k)} - \bar{\boldsymbol{\theta}})^\top \tag{4.6}$$

$$\mathrm{Var}(\bar{\boldsymbol{\theta}}) = \bar{\boldsymbol{U}} + \frac{m+1}{m} \boldsymbol{B}. \tag{4.7}$$

### 4.2.1 Multivariate Imputation using Chained Equations

Multivariate Imputation using Chained Equations (MICE) is a method for multiple imputation that is based on Gibbs sampling to draw from the multivariate distribution $P(\boldsymbol{X}_\text{mis} \mid \boldsymbol{X}_\text{obs}, \boldsymbol{Z}, \boldsymbol{Y}, \hat{\boldsymbol{\theta}})$ [298–300]. In contrast to joint modeling [251], where a parametric multivariate density $P(\boldsymbol{X}|\Phi)$ with unknown parameters $\Phi$ is constructed to describe the full data, Gibbs sampling uses conditional distributions $P(\boldsymbol{X}^{(j)} \mid \boldsymbol{X}^{(-j)}, \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}_j)$ for each of the $p$ features, with $\boldsymbol{X}^{(j)}$ being a vector corresponding to the $j$-th column of $\boldsymbol{X}$ and $\boldsymbol{X}^{(-j)}$ being identical to $\boldsymbol{X}$ with the $j$-th column removed ($j = 1, \ldots, p$). An important advantage of MICE is that it simplifies handling data consisting of variables of different type (continuous, ordinal, or categorical), because it does not require specifying a joint distribution across all types, which may be difficult to specify and is unfeasible with high-dimensional data [299]. Moreover, having an imputation model for each variable ensures consistency among variables that are derived from other variables in the data. For instance, there is a well defined relationship between the body mass index and weight and height, which should hold for imputed data too [298].

To create a single imputed dataset, MICE repeatedly draws from a Gibbs sampler. It first imputes missing values by randomly sampling with replacement from the observed data $\boldsymbol{X}_{\text{obs}}$. Next, missing values in the first feature are imputed by constructing an imputation model based on non-missing entries of the first feature ($\boldsymbol{X}_{\text{obs}}^{(1)}$), values for the remaining $p-1$ features ($\boldsymbol{X}_{t-1}^{(-1)}$), values of features without missing values ($\boldsymbol{Z}$), and the dependent variable $\boldsymbol{Y}$, which corresponds to the following draws:

$$\begin{aligned}
\boldsymbol{\theta}_t^{(1)} &\sim P\left(\boldsymbol{\theta}^{(1)} \mid \boldsymbol{X}_{\text{obs}}^{(1)}, \boldsymbol{X}_{t-1}^{(2)}, \ldots, \boldsymbol{X}_{t-1}^{(p)}, \boldsymbol{Z}, \boldsymbol{Y}\right), \\
\boldsymbol{X}_t^{(1)} &\sim P\left(\boldsymbol{X}_{\text{mis}}^{(1)} \mid \boldsymbol{X}_{\text{obs}}^{(1)}, \boldsymbol{X}_{t-1}^{(2)}, \ldots, \boldsymbol{X}_{t-1}^{(p)}, \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}_t^{(1)}\right),
\end{aligned}$$

where $\boldsymbol{X}_{t-1}^{(j)}$ denotes the values of the $j$-th feature, observed and imputed, from the previous iteration. The same process is repeated for the second feature, where the imputation model is based on non-missing entries of the second feature ($\boldsymbol{X}_{\text{obs}}^{(2)}$), and values of features $1, 3, \ldots, p$, which includes the previously imputed values of the first feature:

$$\begin{aligned}
\boldsymbol{\theta}_t^{(2)} &\sim P\left(\boldsymbol{\theta}^{(2)} \mid \boldsymbol{X}_{\text{obs}}^{(2)}, \boldsymbol{X}_t^{(1)}, \boldsymbol{X}_{t-1}^{(3)}, \ldots, \boldsymbol{X}_{t-1}^{(p)}, \boldsymbol{Z}, \boldsymbol{Y}\right), \\
\boldsymbol{X}_t^{(2)} &\sim P\left(\boldsymbol{X}_{\text{mis}}^{(2)} \mid \boldsymbol{X}_{\text{obs}}^{(2)}, \boldsymbol{X}_t^{(1)}, \boldsymbol{X}_{t-1}^{(3)}, \ldots, \boldsymbol{X}_{t-1}^{(p)}, \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}_t^{(2)}\right).
\end{aligned}$$

The remaining features with missing values are imputed following the same scheme. Thus, the imputation model of the $p$-th feature is based on all previous imputations in this iteration:

$$\begin{aligned}
\boldsymbol{\theta}_t^{(p)} &\sim P\left(\boldsymbol{\theta}^{(p)} \mid \boldsymbol{X}_{\text{obs}}^{(p)}, \boldsymbol{X}_t^{(1)}, \ldots, \boldsymbol{X}_t^{(p-1)}, \boldsymbol{Z}, \boldsymbol{Y}\right), \\
\boldsymbol{X}_t^{(p)} &\sim P\left(\boldsymbol{X}_{\text{mis}}^{(p)} \mid \boldsymbol{X}_{\text{obs}}^{(p)}, \boldsymbol{X}_t^{(1)}, \ldots, \boldsymbol{X}_t^{(p-1)}, \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}_t^{(p)}\right),
\end{aligned}$$

which completes the $t$-th iteration. The process of sequential imputation is repeated multiple times, usually between 5 and 20 times, to yield one imputed dataset [299]. By using different starting values and running $m$ imputation procedures in parallel, one obtains $m$ imputed datasets that can be analyzed with complete data methods. Regarding an adequate number of imputations, White *et al.* [316, p. 388] suggested "the rule of thumb that $m$ should be at least equal to the percentage of incomplete cases."

### 4.2.2 Imputation Models

In the description of the MICE algorithm in the previous section, the imputation model, which is used to impute missing values of the $j$-th feature, remained unspecified. Because MICE imputes values sequentially by iterating over all features, the choice of imputation model depends on the type of the $j$-th feature. Common choices are linear regression or predictive mean matching for continuous values, multinomial logistic regression for categorical features, and the proportional odds model for ordered categorical features [246]. An overview of univariate imputation methods is presented in [298, 316].

The limitation of simple univariate imputation models is that non-linear effects and interactions between features have to be defined manually to obtain unbiased imputations [61, 260]. In addition, these models are unsuitable for high-dimensional data or data with highly correlated features. Classification and regression trees (CART; [39]) offer a suitable alternative, because they can automatically consider interaction effects and their recursive structure allows constructing subgroup-specific imputation models in the leaf nodes. For instance, a tree with two leaf nodes and a split according to gender would use separate imputation models for male and female patients.

Burgette and Reiter [47] proposed to use CART as imputation model for continuous features. For the $j$-th feature, a tree predicting values of the $j$-th feature is trained, using all samples where the $j$-th feature is available. To impute a missing value of one sample, the feature vector consisting of the $p - 1$ other features is dropped down the tree and the imputed value is randomly sampled from the empirical distribution of the $j$-th feature from observed samples that reached the corresponding leaf node. A similar approach can be used to impute categorical missing values by using a classification tree instead of a regression tree.

The idea can also be extended to random forests, which are an ensemble of randomized classification or regression trees [37]. Stekhoven and Bühlmann [275] constructed a random forest to predict values of the $j$-th feature from all samples where the relevant feature was observed. Their imputation procedure simply consists of using the predicted value of the forest without incorporating additional randomness. Shah *et al.* [262] experimentally showed that both the CART approach [47] as well as the random forest approach [275] resulted in biased parameter estimates of Cox's proportional hazards model. Therefore, Shah *et al.* [262] proposed a modification where the imputed value is non-deterministic. For categorical features, they proposed to impute missing values from a forest by randomly picking a tree from the ensemble and using its prediction as imputed value. For continuous values, their imputation consisted of "random draws from independent normal distributions centered on conditional means predicted using random forest" [262, p. 765]. Yet another imputation model based on random forests was proposed by Doove *et al.* [79]. They suggested to first record the leaf nodes of all trees that were reached by a sample with missing $j$-th feature. Next, they constructed the empirical distribution of the $j$-th feature by combining all observed samples from the previously recorded leaf nodes. The imputed value is selected by randomly sampling from the resulting empirical distribution of the $j$-th feature.

### 4.2.3 Multiple Imputation for the Cox Model

White and Royston [315] studied using MICE for survival data when survival time follows Cox's proportional hazards model [67]. They investigated different approaches to incorporate the survival time into an imputation model: survival time as is, log-transformation of survival time, squared transformation of survival time, and an estimate of the cumulative baseline hazard function instead of the survival time. With respect to the latter, they proposed three alternatives to estimate the cumulative baseline hazard function: non-parametricly using the Nelson-Aalen estimator (2.20) or via Cox's proportional hazards model that is updated in each, or every $k$-th iteration of MICE. They suggested to build imputation models based on the event indicator $\delta_i$ and the value of the Nelson-Aalen estimator of the cumulative baseline hazard function $\hat{H}(y_i)$[315].

# 5 Fast Training of Survival Support Vector Machine with Ranking Constraints

I will begin this chapter by having a closer look at existing training algorithms for survival support vector machines presented in section 3.3.1 (page 46) and by highlighting some weaknesses of these approaches. Initially, I will focus on improving training of models with *linear* decision function, which is largely based on work by Lee and Lin [190] and Pölsterl *et al.* [236], until section section 5.6, where I will show that similar ideas can be used to efficiently train *non-linear* models via the kernel trick. Finally, I will demonstrate the advantage of the improved training algorithm on synthetic and real-world data in section 5.7.

## 5.1 Analysis of Existing Approaches

First, let me repeat the definition of the objective function of ranking-based survival support vector machines in eq. (3.48) on page 46:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma \sum_{(i,j)\in\mathcal{P}} \xi_{ij} \\ \text{subject to} \quad & \boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \boldsymbol{x}_j \geq 1 - \xi_{ij}, \quad && \forall (i,j) \in \mathcal{P}, \\ & \xi_{ij} \geq 0, && \forall (i,j) \in \mathcal{P}. \end{aligned}$$

The corresponding dual function (3.74) on page 55 has the form

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \mathbb{1}_m - \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{A}^\top\boldsymbol{\alpha} \\ \text{subject to} \quad & 0 \leq \alpha_{ij} \leq \gamma, \quad \forall (i,j) \in \mathcal{P}, \end{aligned}$$

where $m = |\mathcal{P}|$ and $\boldsymbol{A}$ is a $m \times n$ sparse matrix that encodes comparable pairs in $\mathcal{P} = \{(i,j) \mid y_i > y_j \wedge \delta_j = 1\}_{i=1}^n$.

The main disadvantage of ranking-based techniques is that their objective function (3.48) depends on the size of the set $\mathcal{P}$, which scales quadratically in the number of

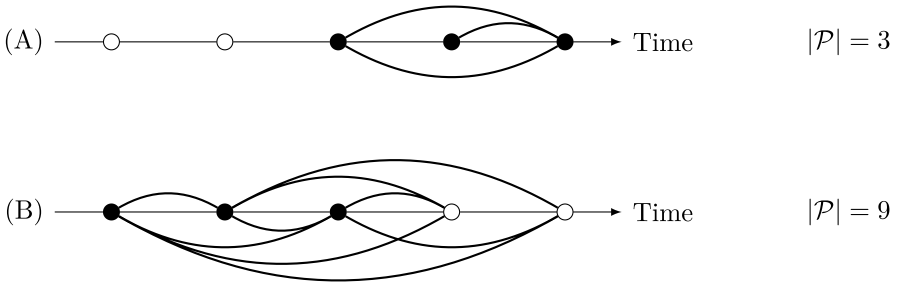


**Figure 5.1**: Comparable pairs between uncensored (black) and censored (white) samples.

training samples. Without censoring, all pairwise comparisons have to be considered during training and the size of $\mathcal{P}$ is $\binom{n}{2} = n(n-1)/2$. Consequently, the optimization problem in eq. (3.48) consists of a quadratic number of constraints with respect to the number of training samples. If part of the survival times are censored, the size of $\mathcal{P}$ depends on the amount of uncensored records and the order of observed time points, censored and uncensored. Let $n_e$ indicate the number of events in a dataset of $n$ samples, then the size of $|\mathcal{P}|$ is at least $n_e(n_e - 1)/2$. This situation arises if all censored subjects drop out before the first event was observed, hence, all uncensored records are incomparable to all censored records (see top of fig. 5.1). If the situation is reversed and the first time point of censoring occurs after the last time point of an observed event, all uncensored records can be compared with all censored records, which means $|\mathcal{P}| = n_e(n_e - 1)/2 + n_e(n - n_e)$ (see bottom of fig. 5.1). By expressing $n_e$ as $n_e = q_e n$, with $q_e$ being the percentage of events, the minimum size of $\mathcal{P}$ is $q_e n(q_e n - 1)/2$ and the maximum size is $q_e n^2 - q_e n(q_e n + 1)/2$. This shows that in both cases $|\mathcal{P}|$ is of the order of $O(q_e n^2)$ and the number of constraints in the optimization problem (3.48) is squared in the number of samples in the presence of right censoring, too.

Evers and Messow [90] and Van Belle *et al.* [294] solved the optimization problem in eq. (3.48) by constructing the Lagrangian dual function in eq. (3.74) and using a standard solver for convex quadratic programming, such as the sequential minimal optimization (SMO) algorithm [234]. The dual function depends on a quadratic number of constraints and requires space in the order of $O(q_e n^2)$ to construct the matrix $\boldsymbol{A}$. Consequently, training requires $O(pq_e^2 n^4)$ time, which is intractable with medium to large sized datasets. In addition, finding a good approximate solution to the dual does not necessarily result in a good solution for the primal, which is of primary interest. Hush *et al.* [154, Theorem 2] proved that in order to obtain a solution to the primal objective function that approximates the optimal primal solution by $\varepsilon_{\text{primal}}$, the error of the approximate dual solution has to be $O(\gamma^{-1}\varepsilon^2_{\text{primal}})$, where $\gamma > 0$ is the regularization parameter that weights the influence of the loss function. Therefore, algorithms optimizing the dual objective function result in a slow convergence rate in the primal objective function.

Authors in [295] tackled the high space and time complexity of ranking-based survival support vector machines by restricting the number of constraints involved in the optimization problem, such that the new optimization problem approximates the original one. They first clustered subjects according to survival time, and only included those constraints that involved the $k$ nearest neighbors in survival time for each individual. Setting $k = 1$ reduces the training time to $O(pq_e^2 n^2)$. This idea was adopted in [297], albeit for a slightly different objective function (cf. section 3.3.1 on page 46).

Although this approach does lower the training time, its solution differs from the one obtained by solving the full optimization problem. Interestingly, in the absence of censoring, a survival support vector machine with ranking constraints is equivalent to Rank SVM [138], for which Airola *et al.* [4] and Lee and Lin [190] proposed the use of order statistics trees to obtain training algorithms with lower time and space complexity than the naïve approach currently used to train a survival support vector machine [90, 294].

In the remainder of this chapter, I will show that the work in [190] can be adapted to efficiently train a survival support vector machine on right censored data with a time and space complexity independent of the size of $\mathcal{P}$, despite considering a quadratic number of constraints in the objective function. The improved optimization scheme uses truncated Newton optimization to minimize the objective function in the primal rather than the dual and avoids explicitly constructing the matrix $\boldsymbol{A}$ by employing order statistics trees.

## 5.2 Survival Support Vector Machine with Squared Hinge Loss

First, I will make a slight change regarding the objective function. The loss function of the unconstrained optimization problem of survival SVM with ranking constraints in eq. (3.49) on page 46 uses the non-differentiable hinge loss. Instead, I will use the differentiable squared hinge loss, which allows employing gradient descent methods to minimize the unconstrained optimization problem. Figure 5.2 illustrates the difference between the loss functions.

> **Definition 5.1.** Given training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^n$ and the set of comparable pairs $\mathcal{P} = \{(i, j) \mid y_i > y_j \land \delta_j = 1\}_{i=1}^n$, the objective function of ranking-based linear survival support vector machine with squared hinge loss is defined as
>
> $$R(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2} \sum_{i,j \in \mathcal{P}} \max(0, 1 - (\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \boldsymbol{x}_j))^2,$$

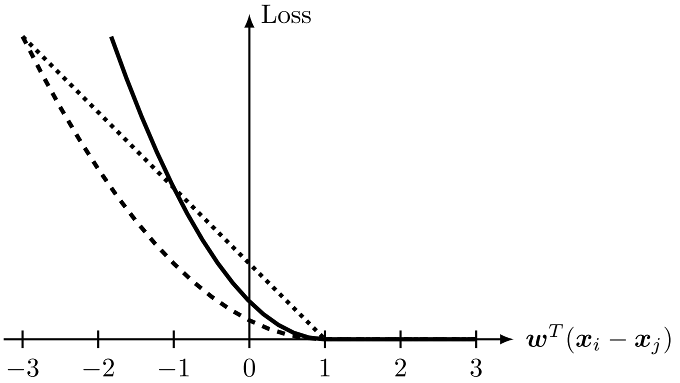


**Figure 5.2**: Comparison of squared hinge loss (solid and dashed line) and hinge loss (dotted line). The solid and dotted line correspond to regularization parameter $\gamma$ set to 1, and the dashed line to $\gamma = 0.5$.

where $\boldsymbol{w} \in \mathbb{R}^p$ are the coefficients and $\gamma > 0$ is a regularization parameter. Alternatively, the objective function can be expressed in matrix form as

$$R(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \frac{\gamma}{2}\left(\mathbb{1}_m - \boldsymbol{A}\boldsymbol{X}\boldsymbol{w}\right)^\top \boldsymbol{D}_{\boldsymbol{w}}\left(\mathbb{1}_m - \boldsymbol{A}\boldsymbol{X}\boldsymbol{w}\right), \tag{5.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a sparse matrix with $\boldsymbol{A}_{k,i} = 1$ and $\boldsymbol{A}_{k,j} = -1$ if $(i, j) \in \mathcal{P}$ and zero otherwise. $\boldsymbol{D}_{\boldsymbol{w}}$ is a $m \times m$ diagonale matrix that has an entry for each $(i, j) \in \mathcal{P}$ that indicates whether this pair is a support vector, i.e., $1 - (\boldsymbol{w}^\top\boldsymbol{x}_i - \boldsymbol{w}^\top\boldsymbol{x}_j) > 0$ [190]. For the $k$-th item of $\mathcal{P}$, representing the pair $(i, j)$, the corresponding entry in $\boldsymbol{D}_{\boldsymbol{w}}$ is defined as

$$(\boldsymbol{D}_{\boldsymbol{w}})_{k,k} = \begin{cases} 1 & \text{if } \boldsymbol{w}^\top\boldsymbol{x}_j > \boldsymbol{w}^\top\boldsymbol{x}_i - 1, \\ 0 & \text{else.} \end{cases} \tag{5.2}$$

The risk score of experiencing an event for a new data point $\boldsymbol{x}_{\text{new}}$ can be estimated by $\hat{f}(\boldsymbol{x}_{\text{new}}) = \hat{\boldsymbol{w}}^\top\boldsymbol{x}_{\text{new}}$ with $\hat{\boldsymbol{w}} = \operatorname{argmin} R(\boldsymbol{w})$.

By using the squared hinge loss, the resulting objective function is differentiable and convex in $\boldsymbol{w}$, which enables the use of Newton's method to minimize it with respect to $\boldsymbol{w}$. One update in Newton's method with step size $\mu$ becomes

$$\boldsymbol{w}^{\text{new}} = \boldsymbol{w} - \mu \left(\frac{\partial^2 R(\boldsymbol{w})}{\partial\boldsymbol{w}\partial\boldsymbol{w}^\top}\right)^{-1} \frac{\partial R(\boldsymbol{w})}{\partial\boldsymbol{w}} \tag{5.3}$$

with partial derivatives

$$\frac{\partial R(\boldsymbol{w})}{\partial\boldsymbol{w}} = \boldsymbol{w} + \gamma\boldsymbol{X}^\top\left(\boldsymbol{A}^\top\boldsymbol{D}_w\boldsymbol{A}\boldsymbol{X}\boldsymbol{w} - \boldsymbol{A}^\top\boldsymbol{D}_w\mathbb{1}_m\right) \tag{5.4}$$

$$\frac{\partial^2 R(\boldsymbol{w})}{\partial\boldsymbol{w}\partial\boldsymbol{w}^\top} = \boldsymbol{I}_p + \gamma\boldsymbol{X}^\top\boldsymbol{A}^\top\boldsymbol{D}_w\boldsymbol{A}\boldsymbol{X}. \tag{5.5}$$

To obtain the second-order derivative, I used the generalized Hessian, because $R(\boldsymbol{w})$ is not twice differentiable at $\boldsymbol{w}$ [170].

Note that the expression $\boldsymbol{A}^\top \boldsymbol{D}_{\boldsymbol{w}} \boldsymbol{A}$ appears in the objective function, its first- and second-order derivative. Multiplying $\boldsymbol{A}^\top$ with the diagonale matrix $\boldsymbol{D}_w$ has the effect that rows not corresponding to support vectors – pairs $(i, j) \in \mathcal{P}$ for which $1 - (\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \boldsymbol{x}_j) < 0$ – are dropped from the matrix $\boldsymbol{A}$. Thus, $\boldsymbol{A}^\top \boldsymbol{D}_{\boldsymbol{w}} \boldsymbol{A}$ can be simplified by expressing it in terms of a new matrix $\boldsymbol{A}_{\boldsymbol{w}} \in \{-1, 0, 1\}^{m_{\boldsymbol{w}}, n}$:

$$\boldsymbol{A}^\top \boldsymbol{D}_{\boldsymbol{w}} \boldsymbol{A} = \boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}},$$

where $m_{\boldsymbol{w}}$ denotes the number of support vectors, which is equivalent to the number of pairs $(i, j) \in \mathcal{P}$ – rows of $\boldsymbol{A}$ – with $\boldsymbol{w}^\top \boldsymbol{x}_j > \boldsymbol{w}^\top \boldsymbol{x}_i - 1$.

**Definition 5.2.** Equation (5.1) can be re-formulated using $\boldsymbol{A}_{\boldsymbol{w}}$ to eliminate $\boldsymbol{D}_{\boldsymbol{w}}$.

$$\begin{aligned} R(\boldsymbol{w}) &= \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2}\left(\mathbb{1}_m^\top \boldsymbol{D}_w - \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{A}^\top \boldsymbol{D}_{\boldsymbol{w}}\right)(\mathbb{1}_m - \boldsymbol{A}\boldsymbol{X}\boldsymbol{w}) \\ &= \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2}\left(m_{\boldsymbol{w}} - 2\boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{A}^\top \boldsymbol{D}\mathbb{1}_m + \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{A}^\top \boldsymbol{D}_w \boldsymbol{A}\boldsymbol{X}\boldsymbol{w}\right) \\ &= \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2}\left(m_{\boldsymbol{w}} + \boldsymbol{w}^\top \boldsymbol{X}^\top \left(\boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_w \boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{A}_{\boldsymbol{w}}^\top \mathbb{1}_{m_w}\right)\right). \end{aligned} \tag{5.6}$$

The corresponding first- and second-order partial derivatives have the form

$$\frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}} = \boldsymbol{w} + \gamma \boldsymbol{X}^\top \left(\boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_w \boldsymbol{X}\boldsymbol{w} - \boldsymbol{A}_{\boldsymbol{w}}^\top \mathbb{1}_{m_w}\right), \tag{5.7}$$

$$\frac{\partial^2 R(\boldsymbol{w})}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} = \boldsymbol{I}_p + \gamma \boldsymbol{X}^\top \boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_w \boldsymbol{X}. \tag{5.8}$$

Although the new formulation is slightly more compact, the main disadvantage from survival support vector machine with hinge loss still applies: The matrix $\boldsymbol{A}_{\boldsymbol{w}}$ has a size of $O(q_e n^2)$, which means computing the Hessian requires $O(m_{\boldsymbol{w}} p^2 + m_{\boldsymbol{w}} n + p)$ operations, rendering training with only a few thousand samples intractable.

## 5.3 Truncated Newton Optimization

Medical research is often challenging due to high-dimensional data: a patient's health record comprises several hundred features, and microarray data consists of several thousand measurements. In these applications, explicitly computing and storing the Hessian matrix constitutes an additional obstacle that makes training prohibitive. Computing and inverting the full Hessian matrix can be avoided when employing a truncated Newton method (see algorithm 5.1) that uses a linear conjugate gradient

**Algorithm 5.1:** Survival Support Vector Machine Training.

```
Input: Training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^{n}$, hyper-parameter $\gamma > 0$.
Output: Coefficients $\boldsymbol{w}$.
Randomly resolve ties in survival times $y_i \; \forall i \in \{1, \ldots, n\}$.
$\boldsymbol{w}^0 \leftarrow \boldsymbol{0}_p$
$t \leftarrow 0$
while not converged do
    Use conjugate gradient to determine search direction $\boldsymbol{u} = \left( \frac{\partial^2 R(\boldsymbol{w})}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} \right)^{-1} \frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}}$ with $\boldsymbol{w} = \boldsymbol{w}^t$
    Choose step size $\mu$ by backtracking line search.
    Update $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t + \mu \boldsymbol{u}$
    $t \leftarrow t + 1$
end
$\boldsymbol{w} \leftarrow \boldsymbol{w}^t$
```

method to compute the search direction [74, 170, 205]. This approach only requires the computation of the Hessian-vector product $\boldsymbol{Hv}$, which can be computed by

$$\boldsymbol{Hv} = \boldsymbol{v} + \gamma \boldsymbol{X}^\top \boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}} \boldsymbol{X} \boldsymbol{v}. \tag{5.9}$$

Thus, the complexity of a single conjugate gradient iteration is $O(np + m_{\boldsymbol{w}} + p)$, when multiplying from the right, which is lower than $O(m_{\boldsymbol{w}} p^2 + m_{\boldsymbol{w}} n + p)$ to obtain the full Hessian matrix.

### 5.3.1 Efficient Calculation of Search Direction

The complexity of a single conjugate gradient iteration to determine the search direction still depends on the matrix $\boldsymbol{A}_{\boldsymbol{w}}$, which has to be recomputed each time $\boldsymbol{w}$ changes, because the set of support vectors might have changed. Constructing $\boldsymbol{A}_{\boldsymbol{w}}$ requires iterating over all comparable pairs, being of order $q_e n^2$. Therefore, the complexity of learning a new model is still quadratic in the number of samples.

Next, I will derive an improved algorithm that avoids constructing $\boldsymbol{A}_{\boldsymbol{w}}$ explicitly. The solution arises by deriving the conditions under which an entry in $\boldsymbol{A}_{\boldsymbol{w}}$ is non-zero, which subsequently suggests a compact representation of an entry in $\boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}}$ and leads to an efficient optimization scheme that is independent of the size of $\mathcal{P}$.

**Proposition 5.1.** *For $k \in \{1, \ldots, m_{\boldsymbol{w}}\}$ and $q \in \{1, \ldots, n\}$, $(\boldsymbol{A}_{\boldsymbol{w}})_{k,q} = 1$ if all the following conditions are satisfied:*

(a) *survival time of q-th sample is* lower *than survival time of some sample $s \in \{1, \ldots, n\}$ (s outlives q): $y_q < y_s$.*

*(b) the $q$-th sample is uncensored: $\delta_q = 1$.*
*(c) the pair $(s, q) \in \mathcal{P}$ is a support vector: $\boldsymbol{w}^\top \boldsymbol{x}_s < \boldsymbol{w}^\top \boldsymbol{x}_q + 1$.*

**Proposition 5.2.** *For $k \in \{1, \dots, m_{\boldsymbol{w}}\}$ and $q \in \{1, \dots, n\}$, $(\boldsymbol{A_w})_{k,q} = -1$ if all the following conditions are satisfied:*

*(a) survival time of $q$-th sample is* higher *than survival time of some sample $s \in \{1, \dots, n\}$ ($q$ outlives $s$): $y_q > y_s$.*
*(b) the $s$-th sample is uncensored: $\delta_s = 1$.*
*(c) the pair $(q, s) \in \mathcal{P}$ is a support vector: $\boldsymbol{w}^\top \boldsymbol{x}_s > \boldsymbol{w}^\top \boldsymbol{x}_q - 1$.*

*Proof.* Note that the only difference between both propositions is the order of samples $s$ and $q$ with respect to their survival times. Thus, the first proposition can be transformed into the second by swaping $s$ and $q$, and vice versa. Conditions (a) and (b) are directly derived from the definition of $\boldsymbol{A}$. Each row of $\boldsymbol{A}$ and $\boldsymbol{A_w}$ contains exactly one element that is 1, one element that is -1, and the rest is all zeros. For each pair of samples (row of $\boldsymbol{A}$), the sample with the shorter survival time is assigned 1, and the other sample -1, which is reflected by condition (a). In addition, each pair must be comparable, i.e., the sample with the shorter survival time must be uncensored, which leads to condition (b). Finally, condition (c) is due to the multiplication $\boldsymbol{AD_w}$ that restricts rows of $\boldsymbol{A}$ to pairs of samples that are support vectors. □

If proposition 5.1 or 5.2 holds, the result of the multiplication $(\boldsymbol{A_w})_{k,i} \cdot (\boldsymbol{A_w})_{k,j}$ is either 1 (if $i = j$) or -1 (if $i \neq j$), for $k \in \{1, \dots, m_{\boldsymbol{w}}\}$ and $i, j \in \{1, \dots, n\}$. In the latter case, the conditions of propositions 5.1 and 5.2 are equivalent. Combining all cases, the product $(\boldsymbol{A_w})_{k,i} \cdot (\boldsymbol{A_w})_{k,j}$ is defined as

1. $(\boldsymbol{A_w})_{k,i} \cdot (\boldsymbol{A_w})_{k,j} = 1$ if $i = j$ and
   a) $(\boldsymbol{A_w})_{k,i} = (\boldsymbol{A_w})_{k,j} = 1$, and proposition 5.1 holds for $q = i$,
   b) or $(\boldsymbol{A_w})_{k,i} = (\boldsymbol{A_w})_{k,j} = -1$, and proposition 5.2 holds for $q = i$,
2. $(\boldsymbol{A_w})_{k,i} \cdot (\boldsymbol{A_w})_{k,j} = -1$ if $i \neq j$ and
   a) $(\boldsymbol{A_w})_{k,i} = 1$, $(\boldsymbol{A_w})_{k,j} = -1$, and proposition 5.1 holds for $q = i, s = j \Leftrightarrow$ proposition 5.2 holds for $q = j, s = i$,
   b) or $(\boldsymbol{A_w})_{k,i} = -1$, $(\boldsymbol{A_w})_{k,j} = 1$, and proposition 5.1 holds for $q = j, s = i$, $\Leftrightarrow$ proposition 5.2 holds for $q = i, s = j$,
3. otherwise $(\boldsymbol{A_w})_{k,i} \cdot (\boldsymbol{A_w})_{k,j} = 0$.

To compactly express the element $\left(\boldsymbol{A_w}^\top \boldsymbol{A_w}\right)_{i,j}$, I define sets $\mathrm{SV}_i^+$ and $\mathrm{SV}_i^-$ that represent propositions 5.1 and 5.2.

$$\mathrm{SV}_i^+ = \{s \mid y_s > y_i \land \boldsymbol{w}^\top \boldsymbol{x}_s < \boldsymbol{w}^\top \boldsymbol{x}_i + 1 \land \delta_i = 1\} \qquad l_i^+ = |\mathrm{SV}_i^+|$$
$$\mathrm{SV}_i^- = \{s \mid y_s < y_i \land \boldsymbol{w}^\top \boldsymbol{x}_s > \boldsymbol{w}^\top \boldsymbol{x}_i - 1 \land \delta_s = 1\} \qquad l_i^- = |\mathrm{SV}_i^-|$$

This allows expressing an entry of $\boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}}$ in the compact form

$$
\begin{aligned}
(\boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}})_{i,j} &= \sum_{k=1}^{m_{\boldsymbol{w}}} (\boldsymbol{A}_{\boldsymbol{w}})_{k,i} (\boldsymbol{A}_{\boldsymbol{w}})_{k,j} \\
&= \begin{cases} l_i^+ + l_i^- & \text{if } i = j, \\ -1 & \text{if } i \neq j, \text{ and } j \in \mathrm{SV}_i^+ \text{ or } j \in \mathrm{SV}_i^-, \\ 0 & \text{else,} \end{cases}
\end{aligned}
\tag{5.10}
$$

where the second case is due to only one addend being non-zero, because each pair of samples is compared only once.

The term $\boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}} \boldsymbol{X} \boldsymbol{v}$ is part of the objective function, its gradient, and the Hessian-vector product. The $i$-th entry of the resulting vector can be computed based on the formulation in eq. (5.10):

$$
\begin{aligned}
(\boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}} \boldsymbol{X} \boldsymbol{v})_i &= (l_i^+ + l_i^-) \boldsymbol{x}_i^\top \boldsymbol{v} - \sum_{s \in \mathrm{SV}_i^+} \boldsymbol{x}_s \boldsymbol{v} - \sum_{s \in \mathrm{SV}_i^-} \boldsymbol{x}_s \boldsymbol{v} \\
&= (l_i^+ + l_i^-) \boldsymbol{x}_i^\top \boldsymbol{v} - \sigma_i^+ - \sigma_i^-,
\end{aligned}
\tag{5.11}
$$

which leads to

$$
\boldsymbol{X}^\top \boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}} \boldsymbol{X} \boldsymbol{v} = \boldsymbol{X}^\top \begin{pmatrix} (l_1^+ + l_1^-) \boldsymbol{x}_1^\top \boldsymbol{v} - (\sigma_1^+ + \sigma_1^-) \\ \vdots \\ (l_n^+ + l_n^-) \boldsymbol{x}_n^\top \boldsymbol{v} - (\sigma_n^+ + \sigma_n^-) \end{pmatrix}. \tag{5.12}
$$

Additionally, the objective function and its gradient contain the term $\boldsymbol{A}_{\boldsymbol{w}}^\top \mathbb{1}_{m_{\boldsymbol{w}}}$, where one component is computed as

$$
\begin{aligned}
(\boldsymbol{A}_{\boldsymbol{w}}^\top \mathbb{1}_{m_{\boldsymbol{w}}})_i &= |\mathrm{SV}_i^+ \cup \mathrm{SV}_i^-| \\
&= |\{(s,t) \mid y_t < y_i < y_s \wedge \delta_t = 1 \wedge \delta_i = 1 \wedge \\
&\qquad \boldsymbol{w}^\top \boldsymbol{x}_s - 1 < \boldsymbol{w}^\top \boldsymbol{x}_i < \boldsymbol{w}^\top \boldsymbol{x}_t + 1\}| \\
&= l_i^- - l_i^+.
\end{aligned}
\tag{5.13}
$$

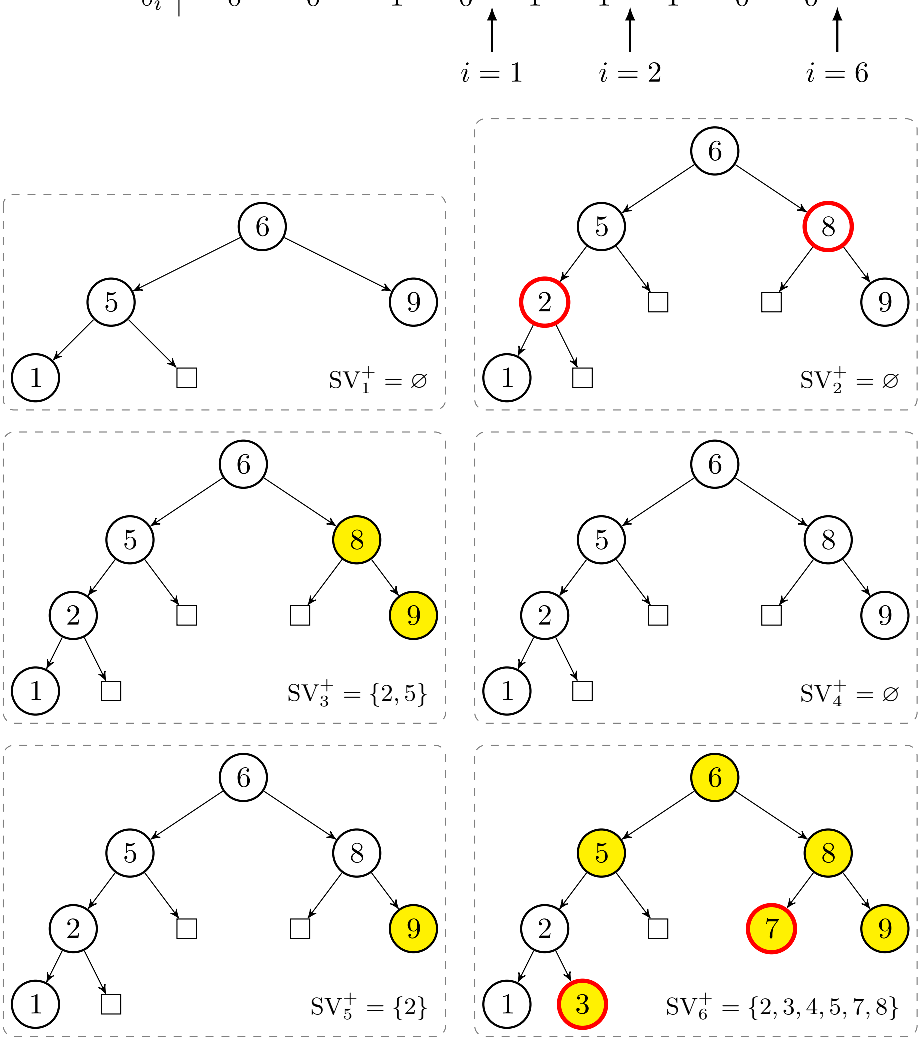
By substituting (5.12) and (5.13) together with $m_{\boldsymbol{w}} = \sum_{i=1}^n l_i^+ = \sum_{i=1}^n l_i^-$ into (5.6), (5.7), and (5.9), all terms that depend on $\boldsymbol{A}_{\boldsymbol{w}}$ during optimization can be eliminated. Assuming $l_i^+$, $l_i^-$, $\sigma_i^+$, and $\sigma_i^-$ have been computed already, the complexity of evaluating the objective function, gradient, and Hessian-vector product is now $O(np + p)$. Subsequently, I will discuss an efficient method to obtain these values using order statistics trees.

## 5.3.2 Improving Optimization by Order Statistics Trees

The main difficulty in constructing the sets $\mathrm{SV}_i^+$ and $\mathrm{SV}_i^-$ stems from the fact that their elements depend on the order of observed time points $y_i$ as well as the order of

**Figure 5.3**: Example illustrating how the set $\mathrm{SV}_i^+$ can be constructed using an order statistics tree. Yellow nodes indicate elements of $\mathrm{SV}_i^+$ and red nodes indicate new elements that have been added to the tree. Arrows below the table indicate the maximum index that is considered when updating a tree, which occurs if $i$ is 1, 2 and 6.

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $\boldsymbol{w}^T\boldsymbol{x}_i$ | -0.7 | -0.1 | 0.15 | 0.2 | 0.3 | 0.8 | 1.6 | 1.7 | 2.3 |
| $y_i$ | 1 | 9 | 6 | 5 | 8 | 2 | 7 | 3 | 4 |
| $\delta_i$ | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |


$i = 1$
$i = 2$
$i = 6$
6
5
9
1
$\mathrm{SV}_1^+ = \varnothing$
6
5
8
2
9
1
$\mathrm{SV}_2^+ = \varnothing$
6
5
8
2
9
1
$\mathrm{SV}_3^+ = \{2, 5\}$
6
5
8
2
9
1
$\mathrm{SV}_4^+ = \varnothing$
6
5
8
2
9
1
$\mathrm{SV}_5^+ = \{2\}$
6
5
8
2
7
9
1
3
$\mathrm{SV}_6^+ = \{2, 3, 4, 5, 7, 8\}$

predictions $\boldsymbol{w}^\top \boldsymbol{x}_i$. I refer to fig. 5.3, which illustrates an exemplary dataset of nine patients and the corresponding values for $\boldsymbol{w}^\top \boldsymbol{x}_i$, $y_i$ and $\delta_i$, to provide some insight on how both sets can be constructed.

**Example 5.1.** First, samples are sorted in ascending order according to $\boldsymbol{w}^\top \boldsymbol{x}_i$, which is already the case for the example in fig. 5.3. Starting from the left, the sets $\mathrm{SV}_1^+$ and $\mathrm{SV}_2^+$ are empty due to both subjects being censored, which violates condition (b) of proposition 5.1. The first non-empty set occurs at $i = 3$ and has two elements: $\mathrm{SV}_3^+ = \{s \mid y_s > 6 \land \boldsymbol{w}^\top \boldsymbol{x}_s < 1.15\} = \{2, 5\}$. The next set ($i = 4$) is again empty, because of censoring, and $\mathrm{SV}_5^+ = \{s \mid y_s > 8 \land \boldsymbol{w}^\top \boldsymbol{x}_s < 1.3\} = \{2\}$.

The example shows, that $\mathrm{SV}_i^+$ is non-empty if and only if the $i$-th sample is uncensored, and that $\mathrm{SV}_{i+1}^+$ can be constructed incrementally from the set $\mathrm{SV}_i^+$:

$$\begin{aligned}\mathrm{SV}_{i+1}^+ &= \{s | \boldsymbol{w}^\top \boldsymbol{x}_s < \boldsymbol{w}^\top \boldsymbol{x}_{i+1} + 1 \land \delta_{i+1} = 1\} \\ &= \{s | \boldsymbol{w}^\top \boldsymbol{x}_s < \boldsymbol{w}^\top \boldsymbol{x}_i + 1\} \cup \{s | \boldsymbol{w}^\top \boldsymbol{x}_i + 1 \leq \boldsymbol{w}^\top \boldsymbol{x}_s < \boldsymbol{w}^\top \boldsymbol{x}_{i+1} + 1 \land \delta_{i+1} = 1\}.\end{aligned}$$

**Example 5.2.** When constructing the set $\mathrm{SV}_i^-$, the first element corresponds to the subject with maximum $\boldsymbol{w}^\top \boldsymbol{x}_i$, i.e., subject 9 in the example in fig. 5.3. Here, $\mathrm{SV}_9^- = \varnothing$, because no element with $\boldsymbol{w}^\top \boldsymbol{x}_s > 1.3$ satisfies conditions (a) and (b) of proposition 5.2. For $i = 8$, $\mathrm{SV}_8^- = \{s \mid y_s < 3 \land \boldsymbol{w}^\top \boldsymbol{x}_s > 0.7 \land \delta_s = 1\} = \{6\}$. The sets $\mathrm{SV}_7^-$ and $\mathrm{SV}_6^-$ are again empty because conditions (a) and (b) of proposition 5.2 are violated, and $\mathrm{SV}_5^- = \{s \mid y_s < 8 \land \boldsymbol{w}^\top \boldsymbol{x}_s > -0.7 \land \delta_s = 1\} = \{7, 6, 3\}$.

Again, the example shows that an incremental update rule can be constructed for $\mathrm{SV}_{i-1}^-$:

$$\begin{aligned}\mathrm{SV}_{i-1}^- =& \{s | \boldsymbol{w}^\top \boldsymbol{x}_s > \boldsymbol{w}^\top \boldsymbol{x}_{i-1} - 1 \land \delta_s = 1\} \\ =& \{s | \boldsymbol{w}^\top \boldsymbol{x}_s > \boldsymbol{w}^\top \boldsymbol{x}_i - 1 \land \delta_s = 1\} \\ & \cup \{s | \boldsymbol{w}^\top \boldsymbol{x}_i - 1 \geq \boldsymbol{w}^\top \boldsymbol{x}_s > \boldsymbol{w}^\top \boldsymbol{x}_{i-1} - 1 \land \delta_s = 1\}.\end{aligned}$$

To maintain the respective sets of relevant samples for computing $\mathrm{SV}_i^+$ and $\mathrm{SV}_i^-$, an order statistics tree that sorts samples according to $y_i$ can be used. By storing values of $y_i$ and $\boldsymbol{w}^\top \boldsymbol{x}_i$ in the tree, the quantities $l_i^+$, $l_i^-$, $\sigma_i^+$, and $\sigma_i^-$, which are required to compute the search direction, can be obtained in logarithmic time. Moreover, the update rules outlined above allow incrementally constructing a single tree for each set without the need of constructing one tree per subject from scratch. To obtain $\mathrm{SV}_{i+1}^+$ from $\mathrm{SV}_i^+$, elements in the set $\{s \mid \boldsymbol{w}^\top \boldsymbol{x}_i + 1 \leq \boldsymbol{w}^\top \boldsymbol{x}_s < \boldsymbol{w}^\top \boldsymbol{x}_{i+1} + 1\}$ are added to the tree, and similarly when constructing $\mathrm{SV}_{i-1}^-$ from $\mathrm{SV}_i^-$, elements from $\{s \mid \boldsymbol{w}^\top \boldsymbol{x}_i - 1 \geq \boldsymbol{w}^\top \boldsymbol{x}_s > \boldsymbol{w}^\top \boldsymbol{x}_{i-1} - 1 \land \delta_s = 1\}$ are added. Note that the incremental update of $\mathrm{SV}_{i+1}^-$ is not restricted by censoring, whereas for $\mathrm{SV}_{i-1}^+$ only $y_s$ and $\boldsymbol{w}^\top \boldsymbol{x}_s$ corresponding to uncensored subjects are added. Hence, the order statistics tree to maintain $\mathrm{SV}_i^-$ is restricted to the survival times of *uncensored* samples, whereas the tree to maintain $\mathrm{SV}_i^+$ contains observed time points $y_i$ of *all* samples, disregarding their censoring status.

**Example 5.3.** This idea is illustrated in fig. 5.3, which shows the steps to compute $\mathrm{SV}_i^+$ from an order statistics tree. Initially, the tree is empty and the update at $i = 1$ consists of adding all elements with $\boldsymbol{w}^\top \boldsymbol{x}_i < 0.3$ (indicated by an arrow) to the tree. Although constructing $\mathrm{SV}_1^+$ is trivial due to censoring, relevant elements are added to the tree for future computations. At $i = 2$, $\mathrm{SV}_2^+$ is empty as well, but two new elements with $0.3 \leq \boldsymbol{w}^\top \boldsymbol{x}_i < 1.3$ are added to the tree (indicated by red nodes). Next, the tree does not need to be updated, but the existing tree is used to obtain the set $\{y_s | y_s > 6\}$ in logarithmic time, which is possible due to the special structure of the tree. The set $\mathrm{SV}_4^+$ is again trivial and constructing $\mathrm{SV}_5^+$ only requires searching for $\{y_s | y_s > 8\}$. Finally, before $\mathrm{SV}_6^+$ can be computed, the tree has to be updated by adding elements with $1.3 \leq \boldsymbol{w}^\top \boldsymbol{x}_i < 1.8$. The procedure to construct $\mathrm{SV}_i^-$ is similar, with the only difference that samples are processed in descending order of $y_i$ and only values corresponding to uncensored samples are added.

Next, I will formally define order statistics trees and the algorithm to compute $l_i^+$, $l_i^-$, $\sigma_i^+$, and $\sigma_i^-$.

> **Definition 5.3.** An order statistics tree is a balanced binary search tree that stores key-value pairs and has the following properties.
>
> 1. For an internal node $x$ with left child $\mathrm{left}(x)$ and right child $\mathrm{right}(x)$:
>
> $$\mathrm{key}(\mathrm{left}(x)) \leq \mathrm{key}(x) \leq \mathrm{key}(\mathrm{right}(x)).$$
>
> 2. For $n$ elements in the tree, the height of the tree is limited by $O(\log n)$.
> 3. Each node $x$ in the tree stores two additional attributes “size” and “sum”.
>    a) size denotes the size of the subtree mounted at $x$:
>
> $$\mathrm{size}(x) = \begin{cases} 0 & \text{if } x = \varnothing, \\ \mathrm{size}(\mathrm{left}(x)) + \mathrm{size}(\mathrm{right}(x)) + 1 & \text{else.} \end{cases}$$
>
>    b) sum denotes the sum of all values in the subtree mounted at $x$:
>
> $$\mathrm{sum}(x) = \begin{cases} 0 & \text{if } x = \varnothing, \\ \mathrm{sum}(\mathrm{left}(x)) + \mathrm{sum}(\mathrm{right}(x)) + \mathrm{value}(x) & \text{else.} \end{cases}$$
>
> 4. The correct value for above attributes is maintained after insertion.

Based on aforementioned definitions, algorithm 5.2 computes $l_i^+$, $\sigma_i^+$, $l_i^-$ and $\sigma_i^-$, where the auxiliary function `CountSmaller` is defined in algorithm 5.3, and `CountLarger`

**Algorithm 5.2:** Efficient computation of $l_i^+$, $l_i^-$, $\sigma_i^+$, and $\sigma_i^-$.

```
Input: Training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \delta_i)\}_{i=1}^n$, coefficient vectors $\boldsymbol{w}$ and $\boldsymbol{v}$.
Output: $l_i^+$, $l_i^-$, $\sigma_i^+$, and $\sigma_i^-$ $\forall i \in \{1, \ldots, n\}$.
Sort all $\boldsymbol{w}^\top \boldsymbol{x}_i$ in ascending order, such that $\boldsymbol{w}^\top \boldsymbol{x}_{\pi(1)} \leq \cdots \leq \boldsymbol{w}^\top \boldsymbol{x}_{\pi(n)}$.
$T \leftarrow$ an empty order statistics tree
$j \leftarrow 1$
for $i \leftarrow 1$ to $n$ do
    while $j \leq n$ and $\boldsymbol{w}^\top \boldsymbol{x}_{\pi(j)} < \boldsymbol{w}^\top \boldsymbol{x}_{\pi(i)} + 1$ do
        Insert $(y_{\pi(j)}, \boldsymbol{x}_{\pi(j)}^\top \boldsymbol{v})$ into $T$
        $j \leftarrow j + 1$
    end
    if $\delta_{\pi(i)} = 1$ then
        $(l_{\pi(i)}^+, \sigma_{\pi(i)}^+) \leftarrow$ CountLarger(root of $T$, $y_{\pi(i)}$)
    else
        $(l_{\pi(i)}^+, \sigma_{\pi(i)}^+) \leftarrow (0, 0)$
    end
end
$j \leftarrow n$
$T \leftarrow$ an empty order statistic tree
for $i \leftarrow n$ to 1 do
    while $j \geq 1$ and $\boldsymbol{w}^\top \boldsymbol{x}_{\pi(j)} > \boldsymbol{w}^\top \boldsymbol{x}_{\pi(i)} - 1$ do
        if $\delta_{\pi(j)} = 1$ then Insert $(y_{\pi(j)}, \boldsymbol{x}_{\pi(j)}^\top \boldsymbol{v})$ into $T$
        $j \leftarrow j - 1$
    end
    $(l_{\pi(i)}^-, \sigma_{\pi(i)}^-) \leftarrow$ CountSmaller(root of $T$, $y_{\pi(i)}$)
end
```

works in a similar manner. The complexity of these functions corresponds to the complexity of finding an element in a binary search tree, which is $O(\log n)$. Hence, the overall complexity of algorithm 5.2 is $O(n \log n)$, and the Hessian-vector product in (5.9) can be carried out in $O(np + p + n \log n)$, after sorting according to $\boldsymbol{w}^\top \boldsymbol{x}_i$, which costs $O(n \log n)$. Thus, one conjugate gradient iteration does not depend on the size of the set of comparable pairs $\mathcal{P}$ anymore, which scales quadratically in the number of samples. Finally, the overall complexity of training a ranking-based survival support vector machine as outlined in algorithm 5.1 is

$$[O(n \log n) + O(np + p + n \log n)] \cdot \bar{N}_{\text{CG}} \cdot N_{\text{Newton}}, \tag{5.14}$$

where $\bar{N}_{\text{CG}}$ and $N_{\text{Newton}}$ are the average number of conjugate gradient iterations and the total number of Newton updates, respectively.

**Algorithm 5.3:** CountSmaller

```
Function CountSmaller($x$, $y_i$)
    Input: node $x$ in order statistics tree, survival time $y_i$.
    Output: $l_i^-$ (number of uncensored samples with $y_s < y_i$), and $\sigma_i^- = \sum_{s \in \mathrm{SV}_i^-} \boldsymbol{x}_i^\top \boldsymbol{v}$.

    if $x = \varnothing$ then
        $l_i^- \leftarrow 0; \sigma_i^- \leftarrow 0$
    else if $\mathrm{key}(x) = y_i$ then
        $l_i^- \leftarrow \mathrm{size}(\mathrm{left}(x))$
        $\sigma_i^- \leftarrow \mathrm{sum}(\mathrm{left}(y))$
    else if $\mathrm{key}(x) < y_i$ then
        $(l_i^-, \sigma_i^-) \leftarrow$ CountSmaller($\mathrm{right}(x), y_i$)
        $l_i^- \leftarrow l_i^- + \mathrm{size}(x) - \mathrm{size}(\mathrm{right}(x))$
        $\sigma_i^- \leftarrow \sigma_i^- + \mathrm{sum}(x) - \mathrm{sum}(\mathrm{right}(x))$
    else // $\mathrm{key}(x) > y_i$
        $(l_i^-, \sigma_i^-) \leftarrow$ CountSmaller($\mathrm{left}(x), y_i$)
    end
    return $l_i^-, \sigma_i^-$
end
```

## 5.4 Survival Analysis as Regression Problem

Instead of treating survival analysis as a ranking problem, authors have proposed regression-based approaches using an absolute loss as well [172, 264] (cf. section 3.3.2 on page 48). A regression model, in contrast to a ranking-based model, can predict the exact time of an event. Training algorithms for such a model need to be aware of censored patient records as well. For right censored observations – those who did not experience an event – no information about the correctness of predicted survival times beyond the time of censoring is available. A valid error can only be computed for patients who experienced an event during the study period, or if the predicted survival time is too early, i.e., before the time of censoring. Experiments in [297] revealed that survival models based on $\varepsilon$-insensitive support vector regression worked equally well if the insensitive zone is set to zero. Hence, the regression objective simplifies to an ordinary least square problem with $\ell_2$ penalty and the additional consideration of right censoring.

$$R_{\mathrm{Regr.}}(\boldsymbol{w}, b) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2} \sum_{i=1}^{n} \left(\zeta_{\boldsymbol{w},b}(y_i, \boldsymbol{x}_i, \delta_i)\right)^2, \tag{5.15}$$

$$\zeta_{\boldsymbol{w},b}(y_i, \boldsymbol{x}_i, \delta_i) = \begin{cases} \max(0, y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - b) & \text{if } \delta_i = 0, \\ y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - b & \text{if } \delta_i = 1, \end{cases} \tag{5.16}$$

where $b \in \mathbb{R}$ is the intercept.

Let $\boldsymbol{R}_{\boldsymbol{w},b}$ be a diagonal matrix with the $i$-th element being 1 if $y_i > \boldsymbol{w}^\top \boldsymbol{x}_i + b$ or $\delta_i = 1$, and zero otherwise. The objective function can be expressed in matrix form as

$$R_{\text{Regr.}}(\boldsymbol{w}, b) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - b\mathbb{1}_n\right)^\top \boldsymbol{R}_{\boldsymbol{w},b}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - b\mathbb{1}_n\right). \tag{5.17}$$

The function $R_{\text{Regr.}}$ is differentiable and convex in $\boldsymbol{w}$ and $b$, thus truncated Newton optimization offers an efficient way to minimize it (cf. algorithm 5.1). Its derivatives with respect to the coefficients $\boldsymbol{w}$ have the form

$$\frac{\partial R_{\text{Regr.}}(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} = \boldsymbol{w} + \gamma \boldsymbol{X}^\top \boldsymbol{R}_{\boldsymbol{w},b}\left(\boldsymbol{X}\boldsymbol{w} + b\mathbb{1}_n - \boldsymbol{y}\right) \tag{5.18}$$

$$\frac{\partial^2 R_{\text{Regr.}}(\boldsymbol{w}, b)}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} = \boldsymbol{I}_p + \gamma \boldsymbol{X}^\top \boldsymbol{R}_{\boldsymbol{w},b}\boldsymbol{X}. \tag{5.19}$$

The intercept $b$ constitutes an additional parameter that has to be considered during optimization. Derivatives involving $b$ have are given by

$$\frac{\partial R_{\text{Regr.}}(\boldsymbol{w}, b)}{\partial b} = \gamma \mathbb{1}_n^\top \boldsymbol{R}_{\boldsymbol{w},b}(\boldsymbol{X}\boldsymbol{w} + b\mathbb{1}_n - \boldsymbol{y}) \tag{5.20}$$

$$\frac{\partial^2 R_{\text{Regr.}}(\boldsymbol{w}, b)}{\partial b \partial b} = \gamma \mathbb{1}_n^\top \boldsymbol{R}_{\boldsymbol{w},b}\mathbb{1}_n = \gamma |\text{SV}| \tag{5.21}$$

$$\frac{\partial^2 R_{\text{Regr.}}(\boldsymbol{w}, b)}{\partial b \partial w_k} = \gamma \sum_{i \in \text{SV}} x_{ik} = \gamma \left(\mathbb{1}_n^\top \boldsymbol{R}_{\boldsymbol{w},b}\boldsymbol{X}\right)_k \tag{5.22}$$

where $\text{SV} = \{i \mid y_i > \boldsymbol{w}^\top \boldsymbol{x}_i + b \vee \delta_i = 1\}$. The Hessian-vector product $\boldsymbol{H}\boldsymbol{v}$ can be computed by combining eqs. (5.19), (5.21) and (5.22):

$$\boldsymbol{H}\boldsymbol{v} = \begin{pmatrix} 0 & \boldsymbol{0}_p^\top \\ \boldsymbol{0}_p & \boldsymbol{I}_p \end{pmatrix} \boldsymbol{v} + \gamma \begin{pmatrix} |\text{SV}| & \mathbb{1}_n^\top \boldsymbol{R}_{\boldsymbol{w},b}\boldsymbol{X} \\ \boldsymbol{X}^\top \boldsymbol{R}_{\boldsymbol{w},b}\mathbb{1}_n & \boldsymbol{X}^\top \boldsymbol{R}_{\boldsymbol{w},b}\boldsymbol{X} \end{pmatrix} \boldsymbol{v}. \tag{5.23}$$

## 5.5 Hybrid Model

Due to the convexity of the ranking as well as the regression objective function it is straightforward to create a hybrid model that combines both objectives; its objective function is defined as

$$\begin{aligned} R_{\text{hybrid}}(\boldsymbol{w}, b) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2}\Bigg[\alpha \sum_{i,j \in \mathcal{P}} \max(0, 1 - (\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{w}^\top \boldsymbol{x}_j))^2 \\ + (1 - \alpha)\sum_{i=1}^{n} \left(\zeta_{\boldsymbol{w},b}(y_i, \boldsymbol{x}_i, \delta_i)\right)^2\Bigg]. \end{aligned} \tag{5.24}$$

The hyper-parameter $\alpha \in [0, 1]$ controls the relative weight of the regression and ranking objective. If $\alpha = 1$, it reduces to the ranking objective, and if $\alpha = 0$, to the regression objective. The objective function $R_{\text{hybrid}}$ can be minimized by combining the gradient information from the ranking objective in eq. (5.7) and the regression objective in eqs. (5.18) and (5.20). The resulting gradient of the hybrid model has the following form:

$$\begin{aligned}\nabla R_{\text{hybrid}}(\boldsymbol{w}, b) = \begin{pmatrix} 0 \\ \boldsymbol{w} \end{pmatrix} + \gamma\alpha \begin{pmatrix} 0 \\ \boldsymbol{X}^\top \left( \boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}} \boldsymbol{X} \boldsymbol{w} - \boldsymbol{A}_{\boldsymbol{w}}^\top \mathbb{1}_{m_w} \right) \end{pmatrix} \\ + \gamma(1-\alpha) \begin{pmatrix} \mathbb{1}_n^\top \boldsymbol{R}_{\boldsymbol{w},b} (\boldsymbol{X}\boldsymbol{w} + b\mathbb{1}_n - \boldsymbol{y}) \\ \boldsymbol{X}^\top \boldsymbol{R}_{\boldsymbol{w},b} \left( \boldsymbol{X}\boldsymbol{w} + b\mathbb{1}_n - \boldsymbol{y} \right) \end{pmatrix}\end{aligned}$$

Similarly, the Hessian-vector product with respect to the hybrid model reveals itself by combining eqs. (5.9) and (5.23):

$$\begin{aligned}\boldsymbol{H}\boldsymbol{v} = \begin{pmatrix} 0 & \mathbf{0}_p^\top \\ \mathbf{0}_p & \boldsymbol{I}_p \end{pmatrix} \boldsymbol{v} + \gamma\alpha \begin{pmatrix} 0 & \mathbf{0}_p^\top \\ \mathbf{0}_p & \boldsymbol{X}^\top \boldsymbol{A}_{\boldsymbol{w}}^\top \boldsymbol{A}_{\boldsymbol{w}} \boldsymbol{X} \end{pmatrix} \boldsymbol{v} \\ + \gamma(1-\alpha) \begin{pmatrix} |\text{SV}| & \mathbb{1}_n^\top \boldsymbol{R}_{\boldsymbol{w},b} \boldsymbol{X} \\ \boldsymbol{X}^\top \boldsymbol{R}_{\boldsymbol{w},b} \mathbb{1}_n & \boldsymbol{X}^\top \boldsymbol{R}_{\boldsymbol{w},b} \boldsymbol{X} \end{pmatrix} \boldsymbol{v}.\end{aligned}$$

Since the ranking and regression part are nicely separated in the gradient and Hessian-vector product, the same efficient optimization scheme as presented in section 5.3.2 can be used to compute the second addend concerning the ranking-based loss in the gradient as well as the Hessian-vector product. As a result, training a hybrid model has the same time complexity as training a purely ranking-based model.

## 5.6 Non-linear Extension

### 5.6.1 Ranking-based Model

Pölsterl *et al.* [236] proposed to obtain a non-linear survival model using the same optimization scheme as above, but with training data transformed by Kernel PCA [257]. This approach was originally applied to Rank SVM by Chapelle and Keerthi [54]. They observed that when projecting training samples into a high-dimensional feature space $\mathcal{H}$, it is not necessary to consider the full space $\mathcal{H}$, but merely the subspace $\mathcal{F} \subset \mathcal{H}$ spanned by the $n$ training samples, which has at most $n$ dimensions. Kernel PCA [257] constructs an orthonormal basis of $\mathcal{F}$ and subsequently allows projecting samples into the space $\mathcal{F}$. When using Kernel PCA with a non-linear kernel function and applying the linear survival model to data projected into $\mathcal{F}$, the decision boundary will be non-linear in the original feature space.

Here, I will pursue an alternative approach based on results in [53], which I briefly mentioned in section 3.3.5 (page 56) already. Chapelle [53] showed that linear models based on arbitrary convex loss functions can be extended to non-linear models without performing minimization of the loss in the dual. The idea was picked up by Kuo *et al.* [180], who proposed a non-linear Rank SVM model that directly optimizes the non-linear objective function in the primal. It is natural to also apply this approach to ranking-based survival support vector machines, which I will describe next.

The main idea to obtain a non-linear decision function is that the objective function is reformulated with respect to finding a function $f$ from a reproducing kernel Hilbert space $\mathcal{H}_k$ with associated kernel function $k$:

$$\min_{f \in \mathcal{H}_k} \quad \frac{1}{2} \|f\|^2_{\mathcal{H}_k} + \frac{\gamma}{2} \sum_{(i,j) \in \mathcal{P}} \max(0, 1 - (f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)))^2 \tag{5.25}$$

Using the representer theorem (see section 3.3.5 on page 53), the function $f$ can be expressed as $f(\boldsymbol{z}) = \sum_{i=1}^n \beta_i k(\boldsymbol{x}_i, \boldsymbol{z})$, which results in the following formulation of the objective function:

$$\begin{aligned} &\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \beta_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{\gamma}{2} \sum_{(i,j) \in \mathcal{P}} \max \left( 0, 1 - \left( \sum_{l=1}^n \beta_l k(\boldsymbol{x}_l, \boldsymbol{x}_i) - \sum_{l=1}^n \beta_l k(\boldsymbol{x}_l, \boldsymbol{x}_j) \right) \right)^2 \\ =& \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \beta_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{\gamma}{2} \sum_{(i,j) \in \mathcal{P}} \max \left( 0, 1 - \sum_{l=1}^n \beta_l (k(\boldsymbol{x}_l, \boldsymbol{x}_i) - k(\boldsymbol{x}_l, \boldsymbol{x}_j)) \right)^2 , \end{aligned}$$

where the norm $\|f\|^2_{\mathcal{H}_k}$ can be computed by using the reproducing kernel property $f(\boldsymbol{z}) = \langle f, k(\boldsymbol{z}, \cdot) \rangle$ and $\langle k(\boldsymbol{z}, \cdot), k(\boldsymbol{z}', \cdot) \rangle = k(\boldsymbol{z}, \boldsymbol{z}')$.

With respect to Rank SVM, Yu *et al.* [325] formulated the transition to the non-linear case with the help of the representer theorem as well, but they performed optimization in the dual and altered the objective function to force sparsity in the coefficients $\boldsymbol{\beta}$. In contrast, Kuo *et al.* [180] directly optimized the primal without altering the objective function. I will follow the same approach.

First, I reformulate the objective function in matrix form through the $n \times n$ symmetric positive definite kernel matrix $\boldsymbol{K}$ with entries $\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$:

$$R(\boldsymbol{\beta}) = \frac{1}{2} \boldsymbol{\beta}^\top \boldsymbol{K} \boldsymbol{\beta} + \frac{\gamma}{2} \left( \mathbb{1}_m - \boldsymbol{A} \boldsymbol{K} \boldsymbol{\beta} \right)^\top \boldsymbol{D}_\beta \left( \mathbb{1}_m - \boldsymbol{A} \boldsymbol{K} \boldsymbol{\beta} \right), \tag{5.26}$$

where I used $\boldsymbol{D}_\beta \mathrel{\hat{=}} \boldsymbol{D}_{\boldsymbol{w}}$ to emphasize that the set of support vectors now depends on the coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)^\top$, i.e., the $k$-th diagonal element is defined as

$$(\boldsymbol{D}_\beta)_{k,k} = \begin{cases} 1 & \text{if } f(\boldsymbol{x}_j) > f(\boldsymbol{x}_i) - 1 \Leftrightarrow \boldsymbol{K}_j \boldsymbol{\beta} > \boldsymbol{K}_i \boldsymbol{\beta} - 1, \\ 0 & \text{else,} \end{cases}$$

where $\boldsymbol{K}_i$ denotes the $i$-th row of kernel matrix $\boldsymbol{K}$.

The objective function (5.26) of non-linear ranking-based survival support vector machine is similar to the linear model in eq. (5.1). In fact, $R(\boldsymbol{\beta})$ is differentiable and convex with respect to $\boldsymbol{\beta}$ as well, which allows employing truncated Newton optimization. The first- and second-order derivative have the form

$$\begin{aligned}
\frac{\partial R(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \boldsymbol{K}\boldsymbol{\beta} - \gamma(\boldsymbol{A}\boldsymbol{K})^\top \boldsymbol{D}_{\boldsymbol{\beta}} \left(\mathbb{1}_m - \boldsymbol{A}\boldsymbol{K}\boldsymbol{\beta}\right) \\
&= \boldsymbol{K}\boldsymbol{\beta} + \gamma \boldsymbol{K}^\top \left(\boldsymbol{A}^\top \boldsymbol{D}_{\boldsymbol{\beta}} \boldsymbol{A}\boldsymbol{K}\boldsymbol{\beta} - \boldsymbol{A}^\top \boldsymbol{D}_{\boldsymbol{\beta}} \mathbb{1}_m\right) \\
\frac{\partial^2 R(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} &= \boldsymbol{K} + \gamma \boldsymbol{K}^\top \boldsymbol{A}^\top \boldsymbol{D}_{\boldsymbol{\beta}} \boldsymbol{A}\boldsymbol{K}
\end{aligned}$$

As in the linear case, the product $\boldsymbol{A}^\top \boldsymbol{D}_{\boldsymbol{\beta}} \boldsymbol{A}$ can be expressed more compactly through the matrix $\boldsymbol{A}_{\boldsymbol{\beta}} \triangleq \boldsymbol{A}_{\boldsymbol{w}}$ and $m_{\boldsymbol{\beta}} \triangleq m_{\boldsymbol{w}}$, which yields

$$R(\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\beta}^\top \boldsymbol{K}\boldsymbol{\beta} + \frac{\gamma}{2}\left(m_{\boldsymbol{\beta}} + \boldsymbol{\beta}^\top \boldsymbol{K}\left(\boldsymbol{A}_{\boldsymbol{\beta}}^\top \boldsymbol{A}_{\boldsymbol{\beta}} \boldsymbol{K}\boldsymbol{\beta} - 2\boldsymbol{A}_{\boldsymbol{\beta}}^\top \mathbb{1}_{m_{\boldsymbol{\beta}}}\right)\right) \tag{5.27}$$

$$\frac{\partial R(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \boldsymbol{K}\boldsymbol{\beta} + \gamma \boldsymbol{K}\left(\boldsymbol{A}_{\boldsymbol{\beta}}^\top \boldsymbol{A}_{\boldsymbol{\beta}} \boldsymbol{K}\boldsymbol{\beta} - \boldsymbol{A}_{\boldsymbol{\beta}} \mathbb{1}_{m_{\boldsymbol{\beta}}}\right) \tag{5.28}$$

$$\frac{\partial^2 R(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = \boldsymbol{K} + \gamma \boldsymbol{K} \boldsymbol{A}_{\boldsymbol{\beta}}^\top \boldsymbol{A}_{\boldsymbol{\beta}} \boldsymbol{K}, \tag{5.29}$$

where the generalized Hessian is used in the second derivative, because $R(\boldsymbol{\beta})$ is not twice differentiable at $\boldsymbol{\beta}$ [170].

Notice that gradient and Hessian share properties that were already addressed in section 5.3.2 by employing order statistic trees; most importantly, computing the search direction requires constructing the matrix $\boldsymbol{A}_{\boldsymbol{\beta}}$, which requires $O(q_e n^2)$ space. Thus, the dependency on the matrix $\boldsymbol{A}_{\boldsymbol{\beta}}$ can be removed by using the result in eq. (5.12):

$$\boldsymbol{K}\boldsymbol{A}_{\boldsymbol{\beta}}^\top \boldsymbol{A}_{\boldsymbol{\beta}} \boldsymbol{K}\boldsymbol{v} = \boldsymbol{K}\begin{pmatrix} (l_1^+ + l_1^-)\boldsymbol{K}_1\boldsymbol{v} - (\sigma_1^+ + \sigma_1^-) \\ \vdots \\ (l_n^+ + l_n^-)\boldsymbol{K}_n\boldsymbol{v} - (\sigma_n^+ + \sigma_n^-) \end{pmatrix}, \tag{5.30}$$

where $\sigma_i^+ = \sum_{s \in \mathrm{SV}_i^+} \boldsymbol{K}_s\boldsymbol{v}$, $\sigma_i^- = \sum_{s \in \mathrm{SV}_i^-} \boldsymbol{K}_s\boldsymbol{v}$.

### 5.6.2 Regression-based and Hybrid Model

The regression objective (5.15) can be extended to incorporate a non-linear model following a similar scheme. As in (5.25), the objective function is now minimized with respect to a function $f$ from a reproducing kernel Hilbert space $\mathcal{H}_k$ with associated kernel function $k$:

$$\min_{f \in \mathcal{H}_k, b \in \mathbb{R}} \quad \frac{1}{2} \|f\|_{\mathcal{H}_k}^2 + \frac{\gamma}{2} \sum_{i=1}^{n} \left[ \delta_i (y_i - f(\boldsymbol{x}_i) - b) + (1 - \delta_i) \max(0, y_i - f(\boldsymbol{x}_i) - b) \right]^2$$

The main difference to the ranking-based model is that the regression model contains an additional, unpenalized intercept term, which is not part of the function $f$ and has to be considered separately. By applying the representer theorem (see section 3.3.5 on page 53) the objective functions becomes

$$R_{\text{Regr.}}(\boldsymbol{\beta}, b) = \frac{1}{2} \boldsymbol{\beta}^\top \boldsymbol{K} \boldsymbol{\beta} + \frac{\gamma}{2} \left( \boldsymbol{y} - \boldsymbol{K}\boldsymbol{\beta} - b \mathbb{1}_n \right)^\top \boldsymbol{R}_{\beta,b} \left( \boldsymbol{y} - \boldsymbol{K}\boldsymbol{\beta} - b \mathbb{1}_n \right). \tag{5.31}$$

Partial derivatives of (5.31) with respect to the coefficients $\boldsymbol{\beta}$ are given by

$$\frac{\partial R_{\text{Regr.}}(\boldsymbol{\beta}, b)}{\partial \boldsymbol{\beta}} = \boldsymbol{K}\boldsymbol{\beta} + \gamma \boldsymbol{K} \boldsymbol{R}_{\beta,b} (\boldsymbol{K}\boldsymbol{\beta} + b \mathbb{1}_n - \boldsymbol{y}) \tag{5.32}$$

$$\frac{\partial^2 R_{\text{Regr.}}(\boldsymbol{\beta}, b)}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = \boldsymbol{K} + \gamma \boldsymbol{K} \boldsymbol{R}_{\beta,b} \boldsymbol{K}, \tag{5.33}$$

and derivatives involving the intercept $b$ are given by

$$\frac{\partial R_{\text{Regr.}}(\boldsymbol{\beta}, b)}{\partial b} = \gamma \mathbb{1}_n^\top \boldsymbol{R}_{\beta,b} (\boldsymbol{K}\boldsymbol{\beta} + b \mathbb{1}_n - \boldsymbol{y}) \tag{5.34}$$

$$\frac{\partial^2 R_{\text{Regr.}}(\boldsymbol{\beta}, b)}{\partial b \partial b} = \gamma \mathbb{1}_n^\top \boldsymbol{R}_{\beta,b} \mathbb{1}_n = |\text{SV}| \tag{5.35}$$

$$\frac{\partial^2 R_{\text{Regr.}}(\boldsymbol{\beta}, b)}{\partial b \partial \beta_k} = \gamma \sum_{i \in \text{SV}} k(\boldsymbol{x}_i, \boldsymbol{x}_k) = \gamma \left( \mathbb{1}_n^\top \boldsymbol{R}_{\beta,b} \boldsymbol{K} \right)_k. \tag{5.36}$$

The objective function for a non-linear hybrid model is simply a combination of the ranking loss in eq. (5.26) and the regression loss in eq. (5.31):

$$\begin{aligned} R_{\text{hybrid}}(\boldsymbol{\beta}, b) = \frac{1}{2} \boldsymbol{\beta}^\top \boldsymbol{K} \boldsymbol{\beta} + \frac{\gamma}{2} \Big[ & \alpha \left( \mathbb{1}_m - \boldsymbol{A}\boldsymbol{K}\boldsymbol{\beta} \right)^\top \boldsymbol{D}_\beta \left( \mathbb{1}_m - \boldsymbol{A}\boldsymbol{K}\boldsymbol{\beta} \right) \\ & + (1 - \alpha) \left( \boldsymbol{y} - \boldsymbol{K}\boldsymbol{\beta} - b \mathbb{1}_n \right)^\top \boldsymbol{R}_{\beta,b} \left( \boldsymbol{y} - \boldsymbol{K}\boldsymbol{\beta} - b \mathbb{1}_n \right) \Big]. \end{aligned} \tag{5.37}$$

Minimization of $R_{\text{hybrid}}(\boldsymbol{\beta}, b)$ can be carried out via truncated Newton optimization by combining the first- and second-order derivatives of non-linear ranking and non-linear regression losses, similar to the linear hybrid model in section 5.5.

### 5.6.3 Complexity

The main difference to the optimization scheme of the linear model is the requirement to store the $n \times n$ kernel matrix $\boldsymbol{K}$. If $\boldsymbol{K}$ cannot be stored in memory, computing the product $\boldsymbol{K}_i\boldsymbol{v}$ requires $n$ evaluations of the kernel function and $n$ operations to compute the product. If evaluating the kernel function costs $O(p)$, the overall complexity is $O(n^2p)$. Thus, computing the Hessian-vector product in the non-linear case consists of three steps, which have the following complexities:

1. $O(n^3p)$ to compute $\boldsymbol{K}_i\boldsymbol{v}$ for all $i = 1, \ldots, n$,
2. $O(n \log n)$ to sort samples according to values of $\boldsymbol{K}_i\boldsymbol{v}$,
3. $O(n^2 + n + n \log n)$ to calculate the Hessian-vector product via (5.30).

This clearly shows that computing the sum over all comparable pairs is no longer the most time consuming task in minimizing the non-linear objective function, as it was when minimizing the linear objective. Instead, computing $\boldsymbol{K}\boldsymbol{v}$ requires much more time and is the dominating factor in minimizing eq. (5.26).

If the number of samples in the training data is small, the kernel matrix can be computed once and stored in memory thereafter, which results in a one-time cost of $O(n^2p)$. It reduces the costs to compute $\boldsymbol{K}\boldsymbol{v}$ to $O(n^2)$ and the remaining costs remain the same. Although pre-computing the kernel matrix is an improvement, computing $\boldsymbol{K}\boldsymbol{v}$ in each conjugate gradient iteration remains the bottleneck. The overall complexity of training a non-linear ranking-based survival support vector machine with truncated Newton optimization and order statistics trees is

$$O(n^2p) + \Big[O(n \log n) + O(n^2 + n + n \log n)\Big] \cdot \bar{N}_{\text{CG}} \cdot N_{\text{Newton}}. \tag{5.38}$$

Note that direct optimization of the non-linear objective function is preferred over the approach in [236], where Kernel PCA was used to transform the data before training, because the complexity of Kernel PCA is $O(n^2p)$ to construct the kernel matrix and $O(n^3)$ to perform singular value decomposition.

Finally, Kuo *et al.* [180] observed that minimizing eq. (5.26) for large sample size is feasible by splitting the whole training data into a number of blocks such that the overall minimization problem can be split into smaller subproblems. Therefore, instead of maintaining one large kernel matrix, it is split into several blocks, which are small enough to remain in memory, and each subproblem is addressed individually.

## 5.7 Comparison of Survival Support Vector Machines

### 5.7.1 Datasets

#### Synthetic Data

Synthetic survival data of varying size was generated following [21]. Each dataset consisted of one uniformly distributed feature in the interval $[18; 89]$, denoting age, one binary variable denoting sex, drawn from a binomial distribution with probability 0.5, and a categorical variable with three equally distributed levels. In addition, ten numeric features were sampled from a multivariate normal distribution $\mathcal{N}_{10}(\boldsymbol{\mu}, \boldsymbol{I}_{10})$ with mean $\boldsymbol{\mu} = (0, 0, 0.3, 0.15, 0.8, 0.67, 0.2, 0, 0.12, 0.3)^\top$. Survival times $t_i$ were drawn from a Weibull distribution with $k = 1$ (constant hazard rate) and $\lambda = 0.9$ according to the formula presented in [21]:

$$t_i = \left( \frac{-\log u_i}{\lambda \exp(f(\boldsymbol{x}_i))} \right)^{1/k},$$

where $u_i$ is uniformly distributed within $[0; 1]$, $f(\cdot)$ denotes a linear or non-linear model that relates the features to the survival time (see below), and $\boldsymbol{x}_i$ is the fourteen-dimensional feature vector of the $i$-th subject. The censoring time $c_i$ was drawn from a uniform distribution in the interval $[0; \tau]$, where $\tau$ was chosen such that about 20% of survival times were censored.

This data generation scheme was used to generated 100 pairs of train and test data of 1,500 samples from either a linear or a non-linear model $f(\cdot)$. Survival times in the test data were not subject to censoring to eliminate the effect of censoring on estimating the performance.

The *linear model* was defined as

$$\begin{aligned} f(\boldsymbol{x}) = 0.05x_{\text{age}} + 0.8x_{\text{sex}} + 0.03x_{\text{C2}} + 0.3x_{\text{C3}} - 0.1x_{\text{N2}} + 0.6x_{\text{N3}} + x_{\text{N5}} \\ - 0.9x_{\text{N6}} + 0.09x_{\text{N8}} + 0.03x_{\text{N9}} + 0.3x_{\text{N10}}, \quad (5.39) \end{aligned}$$

where C2 and C3 indicate the coefficients corresponding to dummy codes of a categorical feature with three categories and N1 to N10 to continuous features sampled from a multivariate normal distribution. Note that the first, fourth and seventh numeric feature are associated with a zero coefficient, thus do not affect the survival time. Multiple datasets were generated by multiplying the coefficients by a random scaling factor uniformly drawn from $[-1; 1]$.

The *non-linear model* consisted of the same coefficients, but combined features non-linearly:

$$\begin{aligned} f(\boldsymbol{x}) = 0.05x_{\text{age}} + 0.8x_{\text{sex}} + 0.03x_{\text{N1}}^2 + 0.3x_{\text{N2}}^{-2} - 0.1x_{\text{N7}} + 0.6x_{\text{N4}}/x_{\text{N2}} \\ + x_{\text{N1}}/x_{\text{N8}} - 0.9\tanh(x_{\text{N6}})/x_{\text{N5}} + 0.09x_{\text{C1}}/x_{\text{sex}} + 0.03x_{\text{C2}}/x_{\text{sex}} + 0.3x_{\text{C3}}/x_{\text{sex}}. \quad (5.40) \end{aligned}$$

### Real Data

In the second set of experiments the focus was on determining the performance of methods on six real-world datasets of varying size, number of features, and amount of censoring (see table 5.1). The Framingham Offspring and the coronary artery disease

**Table 5.1**: Overview of datasets used in comparison of ranking-based survival support vector machines.

| Dataset | $n$ | $p$ | Events | Outcome |
|---|---|---|---|---|
| AIDS study [146] | $1,151$ | 11 | 96 (8.3%) | AIDS defining event |
| | — | — | 26 (2.3%) | Death |
| Breast cancer [77] | 198 | 80 | 51 (25.8%) | Distant metastases |
| Coronary artery disease [217] | $1,106$ | 56 | 149 (13.5%) | Myocardial infarction or death |
| Framingham Offspring [166] | $4,892$ | 150 | 1,166 (23.8%) | Coronary vessel disease |
| Veteran's Lung Cancer [165] | 137 | 6 | 128 (93.4%) | Death |
| Worcester Heart Attack Study [146] | 500 | 14 | 215 (43.0%) | Death |

data contained missing values, which were imputed first using multivariate imputation using chained equations with random forest models for imputation of continuous as well as categorical variables (see section 4.2.1 on page 94 and [79, 299]). To ease computational resources for validation and since the missing values problem was not the focus, one multiple imputed dataset was randomly picked and analyzed. Finally, I normalized continuous variables to have zero mean and unit variance.

### 5.7.2 Computational Efficiency

The first set of experiments studies the question how the theoretical analysis of runtime complexities translates to actual training times for datasets of different size. Three minimization schemes of ranking-based linear support vector machines are included in the experiments: 1) the simple formulation in eq. (5.1), 2) the alternative formulation in eq. (5.6), and 3) the efficient proposed formulation in eq. (5.12).[1]

Figure 5.4 shows the lowest training time of ten repetitions in wall time using truncated Newton optimization to minimize the objective function (5.1). The simple and improved optimization scheme failed with more than 20,000 samples because of excessive memory requirements due to explicitly constructing the sparse matrix $\boldsymbol{A}$ and $\boldsymbol{A}_{\boldsymbol{w}}$, respectively. For all datasets, optimization converged after less than 20 iterations. Although $\boldsymbol{A}$ has to be constructed only once for the simple optimization scheme, training time quickly degenerates because it repeatedly has to be multiplied by $\boldsymbol{X}\boldsymbol{w}$, which takes $O(mn)$ time. The improved optimization updates $\boldsymbol{A}_{\boldsymbol{w}}$ after each iteration of Newton's method, but only needs to perform $O(m_{\boldsymbol{w}}n)$ operations when multiplied by $\boldsymbol{X}\boldsymbol{w}$, which results

[1] All experiments were carried out on a machine with Intel® Core™ i7-4600U CPU, 12 GB of RAM and the following Python packages: Python 3.4.3, numpy 1.10.1, scipy 0.16.0, numexpr 2.4.4, cython 0.23.4, cvxopt 1.1.7, and scikit-learn 0.16.1, where numpy, scipy, numexpr and scikit-learn were compiled with support for the Intel® Math Kernel Library 11.1.

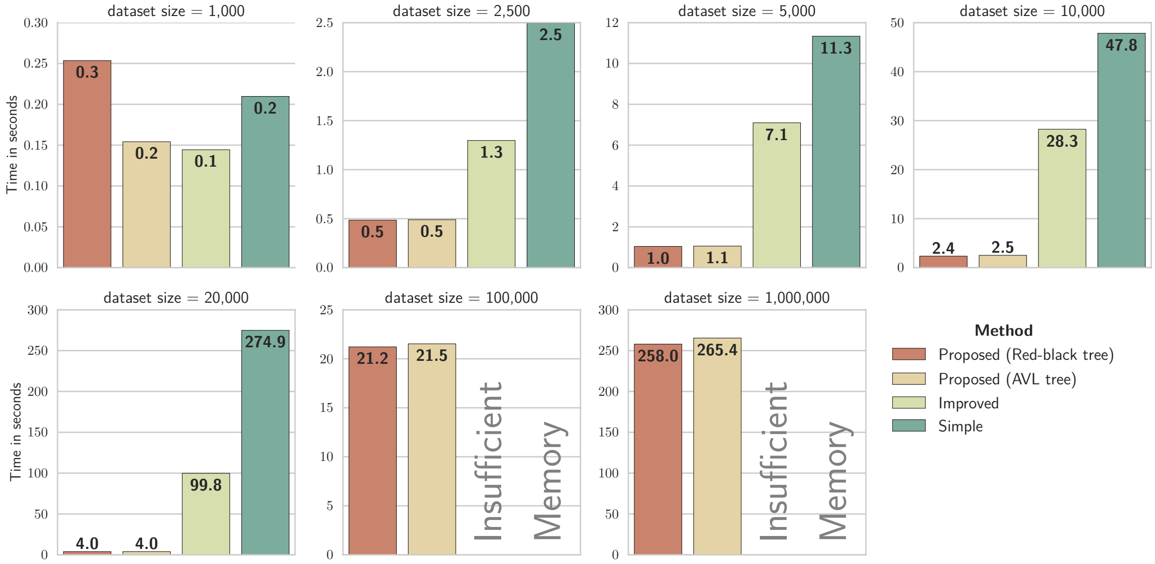


**Figure 5.4**: Training time in seconds of truncated Newton optimization to minimize different formulations of the objective function of linear support vector machines with ranking constraints. *Simple* refers to the objective function in (5.1) and *Improved* to the one in (5.6). *Proposed* refers to the efficient formulation in (5.12) with red-black trees or AVL trees.

in a lower training time. Using order statistics trees, the training time and memory requirements can be lowered significantly; for very large datasets, red-black trees were superior to AVL trees.

### 5.7.3 Prediction Performance

Experiments presented in this section focus on evaluating the predictive performance of ranking-based survival support vector machines on 100 synthetically generated datasets as well as six real-world data sets. The three survival support vector machine models proposed here (ranking, regression, and hybrid) were compared against

1. Cox's proportional hazards model [67] with $\ell_2$ (ridge) penalty (see section 3.2 on page 35),
2. ranking-based survival SVM with hinge loss [90, 294] (see section 3.3.1 on page 46),
3. ranking-based survival SVM with minimum Lipschitz constant [296] (see section 3.3.1 on page 46).

The regularization parameter $\gamma$ for survival SVM controls the weight of the (squared) hinge loss, whereas for Cox's proportional hazards model, $\lambda = \gamma^{-1}$ controls the weight of the $\ell_2$ penalty. Optimal hyper-parameters were determined via grid search by evaluating each configuration using ten random 80%/20% splits of the training data.

The parameters that on average performed best across these ten partitions – according to Harrell's concordance index [127, 128] – were ultimately selected and the model was re-trained on the full training data using optimal parameters. In the grid search, $\gamma$ and $\lambda$ were chosen from the set $\{2^{-12}, 2^{-10}, \ldots, 2^{12}\}$. Similar for $\alpha$, which ranged from 0.05 to 0.95 in steps of 0.05. The maximum number of iterations of Newton's method was two hundred.

### Synthetic Data

The first set of experiments on synthetic data served as a reference on how survival support vector machines compare to each other in a controlled setup. Figure 5.5 summarizes the results on 100 synthetically generated datasets, where all survival times in the test data were uncensored, which leads to unbiased and consistent estimates of the concordance index (see section 3.7 on page 83). In the simple setting, where survival times were generated from a linear model, all the models performed equally well, except for a survival support vector machine with radial basis function (RBF) kernel. Using an RBF kernel to model non-linear relationships was advantageous when data generation was indeed based on a non-linear model, which is evident from the right part of fig. 5.5. The experiments on non-linear synthetic data also revealed that results of a survival support vector machine with RBF kernel were associated with a high degree of instability. On some datasets, using an RBF kernel resulted in a performance increase of up to 0.117 points in concordance index compared to the linear model, whereas on other datasets no increase could be observed. Nevertheless, a survival support vector machine with RBF kernel never performed worse than its linear counterpart: On average, models with RBF kernel achieved a 0.074 points higher concordance index. I also want to mention that the hyper-parameter of the RBF kernel was fixed to its default value ($\sigma^2 = 0.5$) and not optimized during grid search, therefore additional improvements could be possible for the non-linear survival support vector machine.

**Regression.** In addition to evaluating survival support vector machines with respect to concordance of predictions and ground truth, I evaluated the regression-based ($\alpha = 0$) and hybrid model ($\alpha \in \{0.1, 0.2, \ldots, 0.9\}$) in sections 5.4 and 5.5 with respect to the root mean squared error (RMSE) on the test set. An $\ell_2$-penalized accelerated failure time model using inverse probability of censoring weights was used as baseline (see section 3.1.2 on page 34). Figure 5.6 summarizes the results. I observed that median RMSE was relatively high for all three models; it ranged between 273 days for the regression-based objective to 355 days for the accelerated failure time model. Results also show that RMSE was associated with an unusually larger variance. This can be explained by the fact that survival times in the test data were not censored, which resulted in a small portion of outliers with survival times more than ten standard

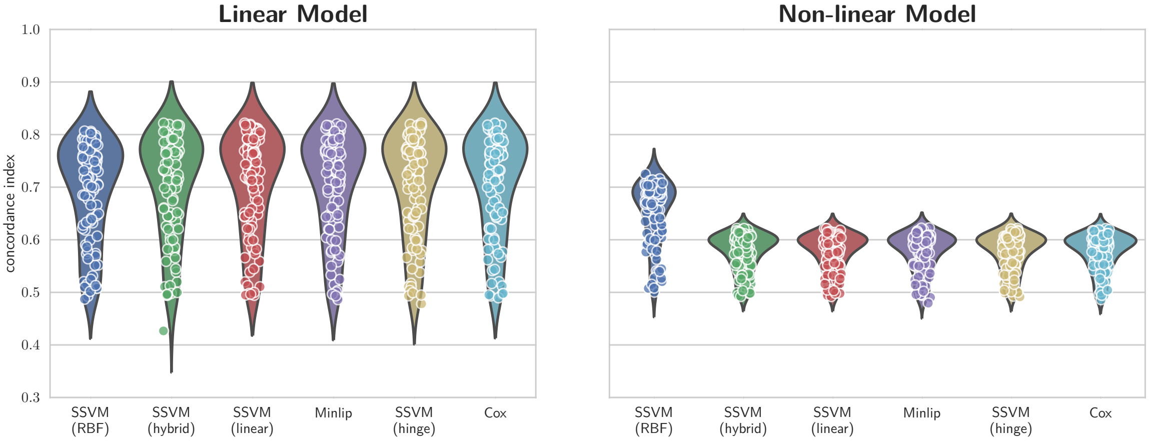


**Figure 5.5**: Performance of ranking-based survival support vector machines and Cox's proportional hazards model on 100 synthetically generated datasets.

deviations from the mean. Naturally, errors with respect to these outliers dominate the RMSE. The concordance index is not affected by outliers, because it only considers the order according to survival time.

### Real Data

In this section, I will discuss results on six real-world datasets using 5-fold cross-validation (including grid search for each fold). In these experiments, test data contained censored samples, which is why performance was measured by Harrell's and Uno's concordance index [127, 292] as well as the integrated area under the time-dependent, cumulative-dynamic ROC curve [153, 293], refer to section 3.7 (page 83) for details. The area under the time-dependent ROC curve was evaluated at time points corresponding to the $10\%, 20\%, \ldots, 80\%$ percentile of the observed time points in the complete dataset. For Uno's concordance index the truncation time was the 80% percentile of the observed time points in the complete dataset.

Results from all experiments are shown in table 5.2. In general, performance measures correlated well and I did not observe any instance were considering a different performance measure would have led to a drastically different result. For the most part, the results on real-world data reflect the outcome of experiments on synthetic data described above. All linear survival support vector machines performed comparably in all but two experiments.

On the breast cancer dataset, models based on minimizing the Lipschitz constant (Minlip) performed worse than models that minimize the (squared) hinge loss. In addition to the difference in loss function, the Minlip model does not consider *all* pairwise relationships between subjects to learn a ranking-based survival model – it only considers the nearest uncensored sample with smaller survival time. The breast

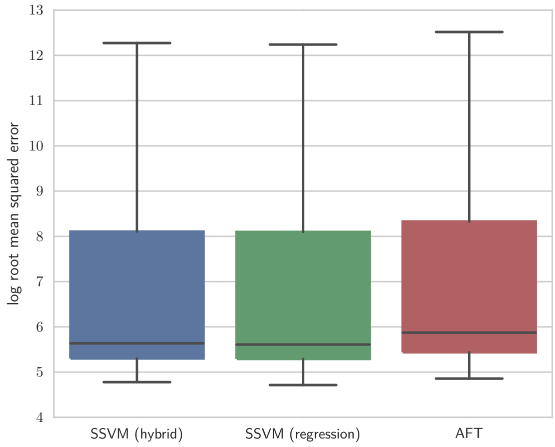


**Figure 5.6**: Performance of regression-based and hybrid survival support vector machines and accelerated failure time (AFT) model on 100 synthetically generated datasets. Far outliers are now shown.

cancer dataset is characterized by a relatively small sample size (198 patients) and a high amount of censoring (74.2%). Therefore, training a Minlip model is based on a set of constraints that is much smaller than what is available to models using the (squared) hinge loss, which ultimately leads to a model that generalizes badly. This result indicates that the proposed efficient optimization scheme can leverage all comparable pairs in the data and lead to a superior predictive survival model.

Interestingly, the relation is reversed when considering results on the coronary artery disease data, where Minlip outperformed the other models. Although the amount of censoring was even higher (86.5%), it contained a larger number of patients such that the total number of comparable pairs in the Minlip model was probably sufficient. Moreover, using an RBF kernel or hybrid model performed better than a linear ranking-based model, but did not reach the level of performance observed for the Minlip model. A possible explanation could be that the data is particularly noisy and reducing the number of constraints in the optimization problem leads to better generalization.

When considering the comparison of linear survival support vector machine models to Cox's proportional hazards model, experiments on three datasets stand out. Linear survival support vector machines outperformed Cox models on the breast cancer dataset, whereas Cox models outperformed all other models on the coronary artery disease data. Results from the AIDS study with death as outcome revealed a weaker advantage of Cox's proportional hazards model: despite the increase in performance in Harrell's and Uno's $c$ index, the difference in the integrated area under the ROC curve was only minor.

With respect to the hybrid model, the results indicate it performed similar to a purely ranking-based model, except for the breast cancer and coronary artery disease data, where the hybrid model performed worse and better, respectively. I could not reproduce

**Table 5.2**: Results on six real-world datasets, reported as the average performance across all folds of 5-fold cross-validation. iAUC: integrated area under the time-dependent, cumulative-dynamic ROC curve.

| | | SSVM<br>(RBF) | SSVM<br>(hybrid) | SSVM<br>(linear) | Minlip<br>(linear) | SSVM<br>(hinge) | Cox<br>(linear) |
|---|---|---|---|---|---|---|---|
| AIDS study (death) | Harrell's $c$ | 0.746 | 0.773 | 0.775 | 0.775 | 0.773 | 0.784 |
| | Uno's $c$ | 0.736 | 0.766 | 0.763 | 0.759 | 0.762 | 0.777 |
| | iAUC | 0.771 | 0.774 | 0.802 | 0.787 | 0.801 | 0.805 |
| AIDS study (AIDS) | Harrell's $c$ | 0.759 | 0.769 | 0.767 | 0.771 | 0.770 | 0.770 |
| | Uno's $c$ | 0.752 | 0.759 | 0.759 | 0.763 | 0.764 | 0.764 |
| | iAUC | 0.759 | 0.796 | 0.766 | 0.774 | 0.771 | 0.771 |
| Breast cancer | Harrell's $c$ | 0.652 | 0.632 | 0.663 | 0.633 | 0.661 | 0.654 |
| | Uno's $c$ | 0.632 | 0.621 | 0.650 | 0.619 | 0.650 | 0.644 |
| | iAUC | 0.690 | 0.639 | 0.682 | 0.641 | 0.674 | 0.662 |
| Coronary artery disease | Harrell's $c$ | 0.739 | 0.733 | 0.706 | 0.756 | 0.714 | 0.768 |
| | Uno's $c$ | 0.745 | 0.735 | 0.708 | 0.755 | 0.718 | 0.760 |
| | iAUC | 0.753 | 0.743 | 0.716 | 0.769 | 0.725 | 0.777 |
| Framingham Offspring | Harrell's $c$ | 0.778 | 0.782 | 0.780 | 0.780 | 0.780 | 0.785 |
| | Uno's $c$ | 0.732 | 0.706 | 0.699 | 0.730 | 0.699 | 0.742 |
| | iAUC | 0.827 | 0.831 | 0.829 | 0.829 | 0.829 | 0.832 |
| Veteran's | Harrell's $c$ | 0.676 | 0.714 | 0.716 | 0.707 | 0.713 | 0.716 |
| | Uno's $c$ | 0.668 | 0.710 | 0.711 | 0.700 | 0.709 | 0.713 |
| | iAUC | 0.740 | 0.781 | 0.783 | 0.777 | 0.783 | 0.780 |
| WHAS | Harrell's $c$ | 0.768 | 0.767 | 0.770 | 0.766 | 0.772 | 0.770 |
| | Uno's $c$ | 0.773 | 0.771 | 0.775 | 0.771 | 0.776 | 0.773 |
| | iAUC | 0.799 | 0.792 | 0.796 | 0.794 | 0.798 | 0.796 |

the results in [297], where hybrid models outperformed ranking-based models. The main difference to the hybrid model here and in [297] is that Van Belle *et al.* combined the regular hinge loss of the Minlip model in eq. (3.51) (page 47) for ranking with the absolute loss for regression, which are both less sensitive to outliers than squared hinge loss and squared error. This problem could be alleviated by introducing sample weights to reduce the influence of outliers in the objective function.

For the remaining experiments, all performance measures agreed that linear survival models performed comparably.

### 5.7.4 Conclusion

I proposed an efficient method for training ranking-based and regression-based survival support vector machines. My algorithm accounts for right censoring of patient records and avoids explicitly constructing a matrix of pairwise constraints – quadratic in the number of samples – by using order statistic trees. I experimentally showed that the reduced time and space complexity allow efficient training of survival support vector machines based on millions of patients, which would otherwise not been possible on commodity hardware. In addition to its high efficiency, the algorithm can be easily adapted for training non-linear as well as hybrid ranking and regression survival support vector machines. This opens up the opportunity to build survival models from large sets of medical records to study the impact of particular factors on a disease or to predict patients' survival.

An implementation of the training algorithm for the survival support vector machine presented in this chapter is available at `http://dx.doi.org/10.5281/zenodo.27733`.

# 6 Survival Analysis For High-Dimensional, Heterogeneous Medical Data

Medical data, such as electronic health records, often consist of a large set of heterogeneous variables, collected from different sources, such as demographics, disease history, medication, allergies, biomarkers, medical images, or genetic markers; each of which offers a different partial view on a patient's state. Moreover, statistical properties among aforementioned sources are inherently different: information about a patient's disease history is often obtained in form of a questionnaire, whereas biomarker measurements denote the concentration of metabolites in the blood; the first is categorical, whereas the second is continuous valued. When analyzing such data, researchers and practitioners are confronted with two problems: 1) the curse of dimensionality – the number of samples required to adequately sample the feature space is increasing exponentially in the number of dimensions – and 2) the heterogeneity in features' sources and statistical properties.

In this chapter, I will focus on two general groups of algorithms for dimensionality reduction, namely *feature selection* and *feature extraction*, and investigate how well these algorithms perform in a wide range of scenarios. In the experiments, I will evaluate 10 combinations of feature extraction methods and 8 survival models with and without intrinsic feature selection in the context of survival analysis on three clinical datasets, which provides empirical evidence when algorithms are expected to perform well or poorly.

## 6.1 Dimensionality Reduction Methods

Feature selection methods assign each feature a value of importance, which is used to filter the set of features, whereas feature extraction methods constructs a new set of features by (non-)linearly combining existing features. Next, I will briefly review methods from both groups of algorithms with a focus on medical applications and survival analysis; for a more general overview, see e.g. [120, 248, 302].

## 6.1.1 Feature Selection

### Wrapper Methods

Feature selection methods can be broadly divided into three categories: filter methods, wrapper methods and embedded methods [120, 248]. Early feature selection approaches belong to the group of wrapper methods. In univariate feature selection, a subset of features is chosen by individually assessing the importance of each feature, for instance, via the $p$-value of the Wald statistic of Cox's proportional hazards model [67]. By fitting individual univariate models and computing the corresponding $p$-values, each feature in a dataset can be assigned an importance value. Ultimately, features with a $p$-value below a certain threshold get selected. The obvious disadvantage of this approach is that features are considered independently from each other, which could lead to selecting redundant features. Moreover, by repeatedly applying the Wald-test to determine the importance of features gives rise to the multiple testing problem [76].

Stepwise forward feature selection is a multivariate feature selection method that starts with a single-feature model, which is incrementally extended by one additional feature [175]. Commonly, the feature that improves the model fit the most is selected, based on either the $p$-value of the Wald-test, as in the univariate case, Akaike's information criterion [6], the Bayesian information criterion [259], or the ratio between the likelihood of the current and the extended Cox model [146]. This process is repeated until adding a feature does not improve the model anymore. Similarly, backward selection starts with a model containing all features and at each step removes the least important feature [175]. For both approaches, the number of comparisons is even higher than in the univariate case, and therefore the impact of multiple testing increases [76].

Due to the high number of comparisons required, univariate, forward, and backward selection, are unsuitable when confronted with many variables, especially when the number of features exceeds the number of samples. Therefore, I will focus on embedded feature selection instead.

### Embedded Methods

Shrinkage methods for Cox's proportional hazards model augment its partial likelihood function by penalizing coefficients that deviate from zero (see section 3.2.5 on page 43). Using the $\ell_2$ norm of the coefficients as penalty leads to ridge regression [306], the $\ell_1$ norm yields the Least Absolute Shrinkage and Selection Operator (LASSO) [288], and the weighted sum of the $\ell_1$ and $\ell_2$ norm results in the elastic net penalty [332]. The adaptive LASSO is based on the $\ell_1$ norm, but penalizes large coefficients less than small coefficients to reduce the bias of the LASSO [328]. Fan and Li showed that the Smoothly Clipped Absolute Deviation (SCAD) penalty satisfies "the mathematical conditions for unbiasedness, sparsity, and continuity" [91, p. 1350] and is preferred over the LASSO. Penalized Cox models select features in the training data by shrinking a subset of coefficients towards zero and thus features with non-zero coefficients are selected.

As mentioned in sections 3.4 and 3.6 (pages 60 and 79), ensemble methods can be used for feature selection as well. A random survival forest [159] accounts for high-dimensional data by considering a random subset of features to determine the best split, thus dramatically reducing the computational costs. Alternatively, gradient boosting constructs an additive model of many weak estimators by functional gradient descent [104]. Models based on gradient boosting differ in the overall loss function that is optimized and the choice of base learners. If base learners are suitable for high-dimensional data, the overall ensemble is well adapted to these situations as well. Here, I will focus on the negative log partial likelihood of Cox's proportional hazards model as loss function [239], and randomized regression trees [37, 39] and componentwise least squares [45] as base learners.

## 6.1.2 Feature Extraction

### Singleview Spectral Embedding

Feature selection methods have been well established in survival analysis, but little work investigated the vast amount of feature extraction methods for dimensionality reduction. Many feature extraction methods were originally proposed for computer vision problems, where data often has more than 100,000 features as well as samples. Note that techniques in manifold learning are considered feature extraction methods too.

Most feature extraction methods are based on spectral decomposition and therefore all require the construction of a matrix that encodes global and/or local relations between data points. Principal component analysis (PCA) [147] considers the relationship between samples on a global scale. First, PCA computes an eigenvalue decomposition of the covariance matrix estimated from the training data. The resulting eigenvectors form the basis of a new space, whose dimensionality can be limited by only selecting $d < p$ eigenvectors corresponding to the $d$ largest eigenvalues. The resulting transformation is linear, but can be extended to the non-linear case via the kernel trick, which yields Kernel PCA [257].

When the neighborhood of a sample is only defined locally, a $n \times n$ neighborhood graph has to be constructed that encodes the neighborhood of each sample. A common choice to measure locality is a $k$ nearest neighbor search based on the Euclidean distance between samples. Using the neighborhood graph, the goal is to find a projection of the data to a low-dimensional space that preserves local neighborhoods as defined in the high-dimensional space (the process is also referred to as *low-dimensional embedding*). Laplacian Eigenmaps (LE; [20]) results in a non-linear transformation of the data, whereas Locality Preserving Projections (LPP; [133]) in a linear transformation. In both cases, a low-dimensional representation can be obtained by limiting the number of eigenvectors after spectral decomposition of the (normalized) graph Laplacian associated with the neighborhood graph.

Feature extraction methods mentioned above assume that feature vectors originate from a common vector space and are called *singleview* spectral embedding methods. Thus, singleview algorithms are not aware of distinct sources of information and statistical properties they imply – which vary heavily in the case of medical records.

### Multiview Spectral Embedding

Dimensionality reduction in the presence of multiple independent groups of features with distinct statistical properties (called *views*) has been addressed by *multiview* spectral embedding (MVSE) methods. The earliest work on multiview dimensionality reduction was presented by Long *et al.* [198], where a separate spectral embedding for each view was constructed, followed by finding a global low-dimensional representation that approximates view-specific embeddings. The approach in [150] differs in using a semi-supervised dimensionality reduction method for each view, and in linearly transforming view-specific embeddings into the common representation. Multiview spectral embedding [322] avoids an initial view-specific embedding and applies an objective function that finds a non-linear, "low-dimensional and sufficiently smooth embedding over all views simultaneously" [322, p. 1438] as well as the amount of complementarity among views. They first constructed a low-dimensional representation using Laplacian Eigenmaps [20] for each view independently, followed by a global coordinate alignment to ensure that low-dimensional embeddings in different views are consistent with each other in the global context. A linearization of the objective function used in MVSE was proposed in [194]. A further extension of MVSE takes into account a sample's class label when constructing the neighborhood graph [196]. Thus, only samples of the same class are connected with each other. Grassmannian regularized structured multiview embedding augments the MVSE objective with three additional terms [309]. The first term measures the distance between graph Laplacians to discover disagreement between views, the second term is used to alleviate the trivial solution of the embedding depending on a single view, and the last term is a structured sparsity penalty to achieve better separation between samples of different classes in the low-dimensional embedding.

Yu *et al.* [326] constructed the neighborhood graph in a supervised manner and used a linear transformation of the high-dimensional data resembling [194]. Gui *et al.* [119] augmented the linear MVSE objective of Yu *et al.* [326] with two additional terms. The first term accounts for correlations between any two views, and the second term performs feature selection, such that the embedding depends on a subset of features only. In [124], view-specific low-dimensional representations were constructed, followed by approximating the concatenation of all low-dimensional embeddings by matrix factorization and imposing structured sparsity. An adaption of locally linear embedding (LLE) [244] for multiple views was presented in [263]. In contrast to MVSE, the authors used LLE to construct view-specific low-dimensional embeddings. Authors in [145] utilized a hypergraph instead of pairwise distances to model relationships between samples. They constructed a multiview hypergraph Laplacian matrix, which was decomposed to obtain the low-dimensional representation. Different from the previous approaches, which were based on spectral decomposition, Xie *et al.* [323] extended $t$-distributed stochastic neighbor embedding to the multiview domain.

### 6.1.3 Methods for Medical Applications

Most feature extraction methods described in the previous section were developed for image processing tasks and evaluated with respect to the classification accuracy after dimensionality reduction. Next, I will present related work with respect to the medical domain.

Dimensionality reduction for gene expression data without including additional patient data from other sources and focusing on classification problems was investigated in [18, 225]. Partial Cox regression [191] is an extension of partial least squares to censored survival data; it has been proposed to analyze gene expression data. A modification that is less sensitive to outliers was proposed by Nguyen and Javier [220]. Supervised principal component analysis only uses features that are correlated with survival time when computing principal components [17]. In “pre-conditioning” [228], supervised principal component analysis is first used to obtain a denoised outcome variable, which subsequently replaces the actual outcome when fitting a survival model with embedded feature selection. Perry *et al.* [231] analyzed text data from medical records of pediatric patients. They proposed supervised Laplacian Eigenmaps, which combines Laplacian Eigenmaps with a supervised loss function. Random indexing for dimensionality reduction of electronic health records to predict adverse drug reactions was proposed in [168].

Finally, several authors implemented comparative studies of feature selection and feature extraction methods for survival analysis in the past. A comparison of penalized Cox models with focus on low-dimensional data was presented in [7, 237], where the latter studied gradient boosting methods as well. Regarding applications of survival analysis for microarray data, Benner *et al.* [26] and Ma *et al.* [204] compared penalized Cox models, Schumacher *et al.* [258] analyzed univariate feature selection, partial Cox regression, and the LASSO, and van Wieringen *et al.* [304] studied the performance of penalized Cox models, partial Cox regression, ensemble methods, and supervised principal component analysis. De Bin *et al.* [72] investigated univariate feature selection, forward stepwise selection, the LASSO, and boosting when combining low-dimensional clinical data with high-dimensional omics data.

In contrast to [18, 168, 225, 231], the focus in my work is on survival analysis rather than classification. Moreover, I will not consider the problem of survival analysis applied to data with more features than samples ($p \gg n$), which has been extensively studied in the context of microarray data already [26, 220, 258, 304]. The work presented by De Bin *et al.* [72] is the closest to mine, because they explicitly consider heterogeneous data consisting of low-dimensional clinical predictors and high-dimensional gene expression data. However, they did not include feature extraction methods in their experiments. The primary goal of this work is to study feature extraction methods in the presence of heterogeneous data, i.e., if information is collected from several sources resulting in distinct groups of features and feature vectors that are a mix of real-valued and categorical predictors.

## 6.2 Multiview Spectral Embedding For Survival Analysis

When it comes to survival data, single- and multiview spectral embedding cannot be applied as is. First, constructing a neighborhood graph based on the Euclidean distance between samples is unsuitable for medical records, because feature vectors are a mix of real-valued and discrete variables, which is not considered by the Euclidean distance. Second, features constructed by non-linear embedding techniques are of limited use, because of their lack of interpretability.

When projecting data into a low-dimensional space, the exact relationship to the original features is unknown. Thus, given a model's coefficients derived from a low-dimensional representation of the data, it is impossible to infer the effect of the original features on survival time. Moreover, when predicting survival of a new patient, it is unknown where the associated feature vector lies with respect to an existing low-dimensional manifold, obtained from the training data. Thus, although non-linear feature extraction techniques may improve predictive performance, their result may be of limited use when the objective is to identify factors or biological pathways that are most decisive for diagnosis. In contrast, when using linear instead of non-linear feature extraction methods, the transformation model simplifies and the connection to the original features can be defined exactly.

Multiview spectral embedding methods, linear and non-linear, may constitute a valuable alternative to singleview methods, because they provide a measure of relevance for each group of features (view). Here, a view corresponds to a feature's broader context, such as disease history, allergies, and so forth. In addition, multiview spectral embedding could be advantageous, because it distinguishes between features originating from different views with different properties, as demonstrated by several applications in computer vision outlined in section 6.1.2.

Therefore, my focus in this chapter will be on multiview spectral embedding and how it can be applied to heterogeneous medical records for survival analysis. I will demonstrate that random survival forests provide a meaningful measure of locality when feature vectors are a mix of continuous and categorical features. Moreover, I will formulate two constraints on the neighborhood graph that incorporate censoring and survival times of patients, respectively. To mitigate the out-of-sample problem of non-linear multiview spectral embedding methods, I propose an interpolation approach that incorporates view importance.

### 6.2.1 Singleview Spectral Embedding

The objective of singleview spectral embedding, such as Laplacian Eigenmaps [20], is to find a low-dimensional non-linear embedding $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n]^\top \in \mathbb{R}^{n \times d}$ of data $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times p}$ $(d < p)$, given local neighborhood relations of data points as encoded by the pairwise affinities $\boldsymbol{W}_{i,j}$ between any two samples $i$ and $j$. The affinity matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ represents the neighborhood graph such that $\boldsymbol{W}_{i,j} \neq 0$ if and only if the $i$-th sample is among the $k$ nearest neighbors of $j$, or $j$ is one of the $k$ nearest neighbors of $i$. Thus, the objective function of singleview spectral embedding is defined as

$$\underset{\boldsymbol{Q}}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{n} \| \boldsymbol{q}_i - \boldsymbol{q}_j \|^2 \boldsymbol{W}_{i,j} = \underset{\boldsymbol{Q}}{\operatorname{argmin}} \operatorname{tr}\Big( \boldsymbol{Q}^\top \boldsymbol{L} \boldsymbol{Q} \Big) , \tag{6.1}$$

where $\boldsymbol{L}$ denotes the normalized graph Laplacian, which is derived from the affinity matrix $\boldsymbol{W}$:

$$\boldsymbol{L} = \boldsymbol{I}_n - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}}, \tag{6.2}$$

where $\mathbf{D}$ is a diagonal matrix with elements $\boldsymbol{D}_{i,i} = \sum_{j=1}^{n} \boldsymbol{W}_{i,j}$, i.e., the degree of the $i$-th node. Note that this formulation is usually only valid if there are no isolated vertices in the neighborhood graph $\boldsymbol{W}$, otherwise one obtains separate low-dimensional embeddings for the disjoint parts of the graph.

### 6.2.2 Multiview Spectral Embedding

In the presence of $m$ independent views, it is assumed that the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ arises from the concatenation of $m$ independent views $\boldsymbol{X}^{(k)} \in \mathbb{R}^{n \times p_k}$, with $1 \leq p_k < p$ $\forall k \in \{1, \ldots, m\}$ such that $p = \sum_{k=1}^{m} p_k$. Thus, the objective function of multiview spectral embedding (MVSE) is a linear combination of the objective in (6.1) with non-negative weights $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^\top$:

$$\begin{aligned}
&\underset{\boldsymbol{\alpha}, \boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(m)}}{\operatorname{argmin}} \quad \sum_{k=1}^{m} \alpha_k \sum_{i=1}^{n} \sum_{j=1}^{n} \| \boldsymbol{q}_i^{(k)} - \boldsymbol{q}_j^{(k)} \|^2 \boldsymbol{W}_{i,j}^{(k)} \\
= &\underset{\boldsymbol{\alpha}, \boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(m)}}{\operatorname{argmin}} \quad \sum_{k=1}^{m} \alpha_k \operatorname{tr}\Big( \left(\boldsymbol{Q}^{(k)}\right)^\top \boldsymbol{L}^{(k)} \boldsymbol{Q}^{(k)} \Big),
\end{aligned} \tag{6.3}$$

where $\boldsymbol{W}^{(k)}$, $\boldsymbol{Q}^{(k)}$, and $\boldsymbol{L}^{(k)}$ denote the affinity matrix, low-dimensional embedding and normalized graph Laplacian of the $k$-th view, respectively.

Next, I assume that views complement each other and that features of one view are sufficient to extract a smooth manifold. Instead of finding a low-dimensional embedding for each view separately, the new objective leverages the complementary nature of views by seeking a global embedding that is consistent with all views. Therefore, finding a low-dimensional embedding $\boldsymbol{Q}$ from all $m$ views simultaneously corresponds to the following optimization problem:

$$\underset{\boldsymbol{Q}, \boldsymbol{\alpha}}{\operatorname{argmin}} \quad \sum_{k=1}^{m} \alpha_k \operatorname{tr}\Big( \boldsymbol{Q}^\top \boldsymbol{L}^{(k)} \boldsymbol{Q} \Big). \tag{6.4}$$

However, the minimum would simply correspond to the view $k^*$ with the minimum $\operatorname{tr}\Big( \boldsymbol{Q}^\top \boldsymbol{L}^{(k^*)} \boldsymbol{Q} \Big)$ and all weights set to zero except $\alpha_{k*} = 1$; essentially ignoring those views that offer complementary information. This problem can be alleviated by replacing $\alpha$ with $\alpha^r$ in (6.4), which encourages $\alpha_k$ values to be close to each other, because $\sum_{k=1}^{m} \alpha_k^r$ is minimized by $\alpha_k = m^{-1}\ \forall k \in \{1, \ldots, m\}$ [308]. The hyper-parameter $r > 1$ controls whether weights are distributed equally: a large value results in weights $\alpha_k$ being close to each other, whereas a value close to 1 leads to a single view being selected and the rest being ignored.

The final objective function of MVSE [322] is defined as:

$$\begin{aligned}
&\underset{\boldsymbol{Q}, \boldsymbol{\alpha}}{\operatorname{argmin}} \quad \sum_{k=1}^{m} \alpha_k^r \operatorname{tr}\Big( \boldsymbol{Q}^\top \boldsymbol{L}^{(k)} \boldsymbol{Q} \Big) \\
&\text{subject to} \quad \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}_d, \quad \sum_{k=1}^{m} \alpha_k = 1, \quad \alpha_k \geq 0.
\end{aligned} \tag{6.5}$$

The orthogonality constraint on $\boldsymbol{Q}$ is required to uniquely determine the solution.

For solving (6.5), I employed the optimization scheme proposed in [322] that alternates between finding the best low-dimensional embedding $\boldsymbol{Q}$, while keeping $\boldsymbol{\alpha}$ fixed, and

**Algorithm 6.1:** Multiview spectral embedding algorithm for survival analysis.

```
Input: Training data $\mathcal{D}^{(k)} = \{(\boldsymbol{x}_i^{(k)}, y_i, \delta_i)\}_{i=1}^n$ for each of the $m$ views, number of nearest
       neighbors, dimension $d$ of the low-dimensional embedding, and complementary
       factor $r > 1$.
Output: View-specific weights $\boldsymbol{\alpha}$, low-dimensional embedding $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$.
foreach $k \in \{1, \ldots, m\}$ do
    Train a random survival forest using data $\mathcal{D}^{(k)}$ from the $k$-th view.
    Retrieve (un)constrained sparse affinity matrix $\boldsymbol{W}^{(k)}$ from the trained forest according
    to (6.9), (6.10), or (6.11).
    Compute normalized Laplacian $\boldsymbol{L}^{(k)}$ as in (6.2).
end
Initialize $\alpha_k = m^{-1}$ $\forall k \in \{1, \ldots, m\}$.
repeat
    Construct average normalized Laplacian matrix $\bar{\boldsymbol{L}} = \sum_{k=1}^m \alpha_k^r \boldsymbol{L}^{(k)}$.
    Update low-dimensional embedding $\boldsymbol{Q}$ using (6.6).
    foreach $k \in \{1, \ldots, m\}$ do
        Update view-specific weight $\alpha_k$ according to (6.7).
    end
    Normalize $\boldsymbol{\alpha}$ by setting $\alpha_k = \alpha_k / \sum_{j=1}^m \alpha_j$, $\forall k \in \{1, \ldots, m\}$.
until convergence
Train survival model with low-dimensional embedding $\boldsymbol{Q}$, event indicators $\boldsymbol{\delta}$ and observed
times $\boldsymbol{y}$.
```

finding the best $\boldsymbol{\alpha}$, while keeping $\boldsymbol{Q}$ fixed:

$$\boldsymbol{Q}^* = \underset{\boldsymbol{Q}}{\operatorname{argmin}} \ \operatorname{tr}\Big(\boldsymbol{Q}^\top \bar{\boldsymbol{L}} \boldsymbol{Q}\Big) \quad \text{subject to} \quad \boldsymbol{Q}^\top \mathbf{Q} = \mathbf{I}_d, \tag{6.6}$$

$$\alpha_k^* = \frac{1 / \Big(\operatorname{tr}\Big(\boldsymbol{Q}^\top \boldsymbol{L}^{(k)} \boldsymbol{Q}\Big)\Big)^{\frac{1}{r-1}}}{\sum_{j=1}^m 1 / \left(\operatorname{tr}(\boldsymbol{Q}^\top \boldsymbol{L}^{(j)} \boldsymbol{Q})\right)^{\frac{1}{r-1}}}, \tag{6.7}$$

where $\bar{\boldsymbol{L}} = \sum_{k=1}^m \alpha_k^r \boldsymbol{L}^{(k)}$ in (6.6) is the weighted average of view-specific normalized Laplacian matrices. The solution to (6.6) is given by the $d$ eigenvectors corresponding to the $d$ smallest eigenvalues of $\bar{\boldsymbol{L}}$ [322]. Equations (6.6) and (6.7) are computed repeatably until convergence. The result is the $d$-dimensional embedding $\boldsymbol{Q}$ and the view-specific weights $\boldsymbol{\alpha}$ after the last iteration. Finally, the low-dimensional representation $\boldsymbol{Q}$ substitutes the original feature vectors $\boldsymbol{X}$ when training a survival model. Algorithm 6.1 summarizes all steps of multiview spectral embedding for survival analysis – steps regarding the construction of the neighborhood graph are discussed in detail below.

### 6.2.3 Construction of Neighborhood Graph

In contrast to applications in image processing, biomedical applications are characterized by a highly heterogeneous set of features with varying properties, and the analysis of time until an adverse event occurs (*survival analysis*), rather than the mere occurrence of an event. Both aspects have to be considered when characterizing local neighborhoods.

The first step in solving the objective function (6.5), is to construct a sparse symmetric affinity matrix $\boldsymbol{W}^{(k)}$ for each view, which subsequently is used to form a normalized graph Laplacian $\boldsymbol{L}^{(k)}$ according to (6.2). Traditionally, neighborhoods are defined based on the Euclidean distance between feature vectors, such that two samples $i$ and $j$ are connected if feature vector $\boldsymbol{x}_i$ is among the $k$ nearest neighbors of $\boldsymbol{x}_j$, or vice versa. Next, a Gaussian weighting function (heat kernel) is used to compute the weights of these edges: $\boldsymbol{W}_{i,j} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / 2\sigma^2)$.

However, this approach has several drawbacks. First, it is subject to the curse of dimensionality, because distances between samples vanish in high-dimensional Euclidean space, leading to poorly defined local neighborhoods [27]. In fact, this is a major limitation of all spectral dimensionality reduction techniques that are based on local neighborhoods of samples [22, 23]. Secondly, the Euclidean distance is unsuitable if feature vectors are a mix of real-valued and discrete variables, because it ignores different statistical properties of features.

To alleviate these problems, I used a random survival forest [159] to obtain the affinity matrix for each view. The advantages of this approach are: 1) it can naturally deal with heterogeneous data, 2) it takes into account the survival time when computing similarities, and 3) it is less affected by the curse of dimensionality. Based on affinities derived from a random survival forest, I define three alternatives to construct the neighborhood graph $\boldsymbol{W}^{(k)}$: 1) without any constraints on edges, 2) prohibiting edges between censored and uncensored samples, and 3) constraining edges to individuals with similar survival times.

**Unconstrained Linking.** Instead of using the Euclidean distance, I defined the distance between samples as the number of times two samples coincided in the same leaf node in a random survival forest, trained on data of the $k$-th view: $\mathcal{D}^{(k)} = \{(\boldsymbol{x}_i^{(k)}, y_i, \delta_i)\}_{i=1}^n$. Based on the definition of proximity in a random survival forest in eq. (3.125) on page 82, the distance between samples $i$ and $j$ for the $k$-th view is defined as

$$\begin{aligned} \mathrm{d}_{\mathrm{RSF}}^{(k)}(\boldsymbol{x}_i, \boldsymbol{x}_j) &= 1 - \mathrm{prox}^{(k)}(\boldsymbol{x}_i^{(k)}, \boldsymbol{x}_j^{(k)}) \\ &= 1 - \frac{1}{B} \sum_{b=1}^{B} \sum_{v=1}^{J_b} I(\boldsymbol{x}_i^{(k)} \in \mathcal{I}_{b,v}^{(k)} \wedge \boldsymbol{x}_j^{(k)} \in \mathcal{I}_{b,v}^{(k)}), \end{aligned} \tag{6.8}$$

where $B$ is the total number of trees in the forest, $J_b$ the number of leaf nodes of the $b$-th tree and $\mathcal{I}_{b,v}^{(k)}$ the set of samples that arrived at the $v$-th leaf node of the $b$-th tree.

Finally, an undirected $k$ nearest neighbor graph and the associated sparse symmetric affinity matrix $\boldsymbol{W}^{(k)}$ can be constructed as follows:

$$\begin{aligned}(\boldsymbol{J}^{(k)})_{i,j} &= \begin{cases}\exp\left(-(\mathrm{d}_{\mathrm{RSF}}^{(k)}(\boldsymbol{x}_i, \boldsymbol{x}_j))^2/2\sigma^2\right) & \text{if } j \in k\mathrm{NN}(i), \\ 0 & \text{otherwise,}\end{cases} \\ \boldsymbol{W}^{(k)} &= \frac{1}{2}\left(\left(\boldsymbol{J}^{(k)}\right)^\top + \boldsymbol{J}^{(k)}\right),\end{aligned} \tag{6.9}$$

where the $k$ nearest neighbor search uses $\mathrm{d}_{\mathrm{RSF}}^{(k)}$ as its distance function.

**Constrained Linking by Censoring.** In classification problems, it is believed that samples of different classes do not share the same underlying manifold. Consequently, one adds a constraint to the construction of the neighborhood graph prohibiting connections between samples of different classes. For a binary classification problem, the graph would consist of two components, resulting in two low-dimensional embeddings, one for each class. For instance when comparing diseased to healthy patients, the constraint forces the low-dimensional representations of samples corresponding to diseased patients to be mapped close to each other and far apart from samples corresponding to healthy patients.

In contrast to classification, supervised information in survival analysis consists of the event indicator $\delta_i$ as well as the observed time $y_i$, and each could be used to construct a supervised neighborhood graph. The naïve approach considers the event indicator as a class label and thus the neighborhood graph is split into two disjoint partitions, one denoting patients, who experienced an event ($\delta_i = 1$), and the other denoting patients with censored survival time ($\delta_i = 0$). Accordingly, eq. (6.9) becomes

$$(\boldsymbol{J}^{(k)})_{i,j} = \begin{cases}\exp\left(-(\mathrm{d}_{\mathrm{RSF}}^{(k)}(\boldsymbol{x}_i, \boldsymbol{x}_j))^2/2\sigma^2\right) & \text{if } j \in k\mathrm{NN}(i) \wedge \delta_i = \delta_j, \\ 0 & \text{otherwise.}\end{cases} \tag{6.10}$$

**Constrained Linking by Observed Time.** The previous approach disregards patients' survival time and can lead to implausible neighborhood relations, especially if one assumes random censoring, i.e., censoring is independent of survival time. Moreover, it is undesirable that samples corresponding to two patients with vastly different survival times could be connected to each other. Consider an example based on the coronary artery disease dataset used in the experiments in section 6.3, where physicians studied the time until death after treatment. If a patient dies during or shortly after intervention, this is most likely due to complications during the procedure. In contrast, if death occurs after a patient has been released from the hospital, the cause is most likely not related to the actual procedure anymore [219]. Therefore, a preferable technique to supervised neighborhood graph construction considers survival time rather than censoring. Here, I discretized the observed time $y_i$ by assigning each patient a label according to which percentile she belonged to and used that information to restrict the nearest neighbor search in (6.9). By defining an additional hyper-parameter $\nu \in \mathbb{N}$ that corresponds to the number of percentiles to use, eq. (6.9) can be modified to yield

$$(\boldsymbol{J}^{(k)})_{i,j} = \begin{cases} \exp\left(-(\mathrm{d}^{(k)}_{\mathrm{RSF}}(\boldsymbol{x}_i, \boldsymbol{x}_j))^2 / 2\sigma^2\right) & \text{if } j \in k\mathrm{NN}(i) \wedge \mathrm{perc}_\nu(y_i) = \mathrm{perc}_\nu(y_j), \\ 0 & \text{otherwise,} \end{cases} \tag{6.11}$$

where $\mathrm{perc}_\nu(y)$ returns the percentile $y$ lies in.

### 6.2.4 Applying Survival Model to Unseen Data Points

After training a survival model based on a low-dimensional representation of the original training according to algorithm 6.1, the model should be able to predict survival for previously unseen patients too. Consequently, a new feature vector has to be projected into the low-dimensional space constructed from the training data first. However, algorithm 6.1 performs a non-linear transformation of the input data, which means that the location of a previously unseen sample on the low-dimensional manifold that represents the training data is unknown. I explored two solutions to this issue: 1) converting the non-linear transformation in (6.5) into a linear one, and 2) interpolating between low-dimensional representations of samples in the original training data.

**Linear Spectral Embedding.** Equation (6.5) can be modified by representing a low-dimensional representation $\boldsymbol{Q}$ as a linear transformation $\boldsymbol{U} \in \mathbb{R}^{p \times d}$ of the input data $\boldsymbol{X}$: $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{U}$. Instead of directly minimizing with respect to $\boldsymbol{Q}$, the objective function is minimized with respect to $\boldsymbol{U}$, resulting in the optimization problem

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{U}, \boldsymbol{\alpha}} && \sum_{k=1}^{m} \alpha_k^r \mathrm{tr}\left(\boldsymbol{X}^\top \boldsymbol{U}^\top \boldsymbol{L}^{(k)} \boldsymbol{X} \boldsymbol{U}\right) \\ &\text{subject to} && \boldsymbol{U}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{U} = \boldsymbol{I}_d, \quad \sum_{k=1}^{m} \alpha_k = 1, \quad \alpha_k \geq 0. \end{aligned} \tag{6.12}$$

Training a survival model based on a linear transformation is analogous to algorithm 6.1, except that (6.12) replaces (6.6) in line 9. The low-dimensional representation $\boldsymbol{q}_{\mathrm{new}}$ of a new sample $\boldsymbol{x}_{\mathrm{new}} \in \mathbb{R}^p$ can be obtained by multiplication with the transformation matrix $\boldsymbol{U}$:

$$\boldsymbol{q}_{\mathrm{new}} = \boldsymbol{U}^\top \boldsymbol{x}_{\mathrm{new}}.$$

**Interpolation of Low-Dimensional Representations.** In the second approach, I retained the non-linear transformation and used interpolation to estimate the location of a new sample in embedding space. I defined the low-dimensional representation $\boldsymbol{q}_{\text{new}}$ of a new sample $\boldsymbol{x}_{\text{new}}$ as the weighted average of the low-dimensional representations $\boldsymbol{q}_i$ of training samples:

$$\boldsymbol{q}_{\text{new}} = \sum_{i=1}^{n} w_i \boldsymbol{q}_i \quad \text{subject to} \quad \sum_{i=1}^{n} w_i = 1.$$

The weight $w_i > 0$ denotes the similarity between $\boldsymbol{x}_{\text{new}}$ and the $i$-th training sample, which can be efficiently retrieved from random survival forests that were used to construct view-specific affinity matrices $\boldsymbol{W}^{(k)}$ according to (6.9), (6.10), or (6.11).

In the context of random survival forests, the proximity measure in eq. (3.125) (page 82) provides an efficient way to estimate the similarity between a new point and all $n$ samples of the training data. Consider the $k$-th of $m$ views and the corresponding subset of features $\boldsymbol{x}_{\text{new}}^{(k)} \in \mathbb{R}^{p_k}$. The similarity $w_i^{(k)}$ of the $i$-th training sample to a new sample with respect to the $k$-th view is defined as

$$w_i^{(k)} = \frac{\exp\left(-(\text{prox}^{(k)}(\boldsymbol{x}_{\text{new}}^{(k)}, \boldsymbol{x}_i^{(k)}))^2\right)}{\sum_{j=1}^{n} \exp\left(-(\text{prox}^{(k)}(\boldsymbol{x}_{\text{new}}^{(k)}, \boldsymbol{x}_j^{(k)}))^2\right)}, \qquad \forall i = 1, \ldots, n, \tag{6.13}$$

where $\text{prox}^{(k)}$ denotes the proximity derived from the random survival forest that was used to construct the affinity matrix $\boldsymbol{W}^{(k)}$. Finally, when considering the similarity of $\boldsymbol{x}_{\text{new}}$ to the training data according to all the $m$ views and accounting for varying importances of views, the combined weight has the form

$$w_i = \sum_{k=1}^{m} \alpha_k w_i^{(k)}, \qquad \forall i = 1, \ldots, n. \tag{6.14}$$

Note that Criminisi *et al.* [69] proposed a similar interpolation scheme applicable to manifold forests in the singleview setting.

## 6.3 Evaluation of Feature Extraction and Feature Selection Method

In this section, I will demonstrate the utility of singleview and multiview spectral embedding algorithms for survival analysis. My evaluation is based on comparing the performance of survival models, trained on a low-dimensional representation of the training data, to survival models with and without embedded feature selection. The experiments serve as empirical evidence to answer the following questions:

1. whether multiview spectral embedding is more favorable than singleview spectral embedding,
2. whether non-linear single and multiview spectral embedding algorithms can improve the capabilities of a linear survival model,
3. whether this improvement is comparable to that of a non-linear survival model with embedded feature selection,
4. whether my proposed technique to construct neighborhood graphs from survival data accurately preserves local neighborhoods,
5. and whether view-specific weights of MVSE provide insight into which group of features plays an important role in a particular disease.

Next, I will provide a full list of evaluated methods and justify my choice of methods.

### 6.3.1 Evaluation Setup

In total, experiments comprised 10 feature extraction methods and 8 survival models, 6 of which with embedded feature selection (see tables 6.1 and 6.2). First, I will describe feature extraction methods used in the experiments followed by feature selection methods.

#### Feature Extraction Methods

With respect to multiview spectral embedding (MVSE), I included the non-linear objective function (6.5) and its linear sibling (6.12). Both MVSE algorithms were paired with three different neighborhood graph construction algorithms: 1) without constraining edges in the graph as in (6.9), 2) by constraining edges by samples' censoring indicator according to (6.10), and 3) by constraining edges by observed time following eq. (6.10). Hence, six variations of MVSE were included in the experiments.

To determine how MVSE techniques compare to singleview methods, i.e., those using the concatenation of all views, Principal Component Analysis [147], Kernel PCA [257], Laplacian Eigenmaps [20], and Locality Preserving Projections [133] were included in the experiments. Each of these baseline methods has different properties (see table 6.1) and served a specific purpose in the evaluation of MVSE methods and answering the questions posed above.

Laplacian Eigenmaps (LE) builds a non-linear transformation and served as singleview baseline to non-linear MVSE, whereas Locality Preserving Projections (LPP), which transforms data linearly, was used as baseline to linear MVSE. Together, (non-)linear MVSE, LE and LPP were used to investigate questions 1 to 3. The objective function of all four algorithms tries to preserve the local neighborhood of samples and hence requires that the underlying manifold is densely sampled to avoid holes in the manifold.

**Table 6.1**: Characteristics of feature extraction methods used in the experiments.

| Method | Neighborhood | Transformation | Views |
|---|---|---|---|
| Principal Component Analysis (PCA) | global | linear | single |
| Kernel PCA | global | non-linear | single |
| Locality Preserving Projections (LPP) | local | linear | single |
| Laplacian Eigenmaps (LE) | local | non-linear | single |
| Multiview Spectral Embedding (MVSE) | local | non-linear | multiple |
| Linear Multiview Spectral Embedding | local | linear | multiple |

Principal Component Analysis (PCA) and Kernel PCA consider relationships globally and are less affected by this problem. Therefore, they were used to determine whether local neighborhoods derived from a random survival forest can capture the underlying manifold of the data better or worse than a global approach (question 4). Finally, the comparison of non-linear to linear feature extraction techniques (single- or multiview) allows determining whether the proposed interpolation scheme in section 6.2.4 can accurately determine the location of new samples on the manifold of the training data.

Each feature extraction method mentioned above and in table 6.1 was combined with a survival model without embedded feature selection, which was trained on the low-dimensional representation obtained via one of these methods. This provides a fair comparison to answer question 3 – how well survival models perform after feature extraction compared to survival models with embedded feature selection. For this purpose, and to handle multicollinearity in the data, I chose ranking-based linear survival support vector machine as described in chapter 5 (page 99).

### Feature Selection Methods

Feature selection methods are an alternative to feature extraction methods when dealing with high-dimensional data. In the experiments I considered survival models with embedded feature selection; they are summarized in table 6.2.

I included Cox's proportional hazards model [67], because it is the standard for analyzing time-to-event data (see section 3.2 on page 35). When combined with a ridge penalty it can be applied to datasets with correlated features, but it does not perform feature selection. If a LASSO penalty is used, the coefficients of a subset of features will be set exactly to zero and features corresponding to non-zero coefficients are selected. Moreover, I included two ensemble methods: random survival forest [159] and gradient boosting [105], which are explained in more detail in sections 3.4 and 3.6 (pages 60 and 79), respectively. Gradient boosting was used to minimize the negative log partial likelihood (3.29) of Cox's proportional hazards model [239] with randomized

**Table 6.2**: Characteristics of survival models used in the experiments.

| Method | Type | Feature Selection |
|---|---|---|
| Survival SVM (SSVM) | linear | no |
| Cox's proportional hazards model (ridge) | linear | no |
| Cox's proportional hazards model (LASSO) | linear | yes |
| Gradient Boosting (trees) | non-linear | yes |
| Gradient Boosting (least squares) | linear | yes |
| Random Survival Forest (RSF) | non-linear | yes |

regression trees [37, 39] or componentwise least squares [45] as base learners. The latter is especially well suited for high-dimensional data, because in each gradient boosting iteration it only selects a single feature and behaves similar to the LASSO [42]. To avoid overfitting of gradient boosting models, I considered stochastic gradient boosting [105] and dropout [238]. The former fits each base learner to a different subset of the training data, and the latter only considers a randomly selected subset of previously fitted base learners in each iteration.

### Validation Scheme

The aforementioned dimensionality reduction methods depend on one or more hyper-parameters that might affect their performance. Therefore, the training phase of each algorithm was augmented by a hyper-parameter search. Hyper-parameters were optimized via grid search by evaluating each configuration using ten random 80%/20% splits of the training data. The parameters that on average performed best across these ten partitions were ultimately selected and the model was re-trained on the full training data using optimal parameters. Results are presented as the mean performance of 5-fold cross-validation.

The number of nearest neighbors used to compute the affinity matrix in (6.9) was set to the logarithm of the number of samples in the respective dataset, as suggested in [202]. For the complementary factor $r$ of multiview spectral embedding in (6.5) and (6.12), I chose to sample more densely close to 1, because small values affect the distribution of weights among views more than large values. When constructing the neighborhood graph, I used a random survival forest with 100 trees and fixed the minimum node size to three. The kernel function used in Kernel PCA was the negative exponential of the distance defined in eq. (6.8). A full list of models' hyper-parameters used in the experiments is available in appendix A.1. The performance of all methods was estimated by Harrell's concordance index ($c$ index; see [127] and section 3.7.1 on page 84).

**Table 6.3**: Overview of datasets used in the experiments. Size denotes the number of samples and the number of features of the dataset. Follow-Up refers to the median follow-up time after enrollment. A detailed description of the views and features they comprised is available in appendix A.2.

| Dataset | Outcome | Size | Views | Events | Follow-Up |
|---|---|---|---|---|---|
| Coronary artery disease [217] | Myocardial infarction or death | $1,233 \times 60$ | 5 | 196 (15.9%) | 4.3 years |
| Breast cancer [77] | Distant metastases | $198 \times 80$ | 2 | 62 (31.3%) | 14.0 years |
| Framingham Offspring [166] | Coronary vessel disease | $4,892 \times 150$ | 7 | 1,166 (23.8%) | 34.6 years |

## Datasets

Experimental evaluation of all methods was based on three different clinical datasets listed in table 6.3. The coronary artery disease data consisted of patients who underwent coronary revascularization procedures for treatment of coronary artery disease [217]; the outcome of interest was the composite of death of any cause and myocardial infarction. The dataset comprised five views: demographics and disease history (12 features), laboratory biomarkers (4 features), angiographic measurements (31 features), medications (6 features), and extent of disease (7 features).

The breast cancer dataset was provided by Desmedt *et al.* [77] and consisted of microarray experiments from primary breast tumors and was used to validate the prognositic value of a 76-gene signature.[1] Expression levels of the 76 genes formed one view and demographic information (4 features) a second view. The objective was to predict the development of distant metastases.

The third dataset was based on data from the Framingham Offspring Study [166], which is a cohort study to investigate risk factors and trends in cardiovascular disease over time. Data consisted of seven views: demographics and disease history (24 features), lipid panel (25 features), laboratory biomarkers (32 features), medication (16 features), menopause (6 features), life-style (17 features), and electrocardiography (30 features). The outcome was the presence of coronary vessel disease before December 31, 2007 (time of censoring).

Missing values in the coronary artery disease and Framingham Offspring data were imputed using multivariate imputation using chained equations with random forest models (see [79, 299] and section 4.2.1 on page 94). To ease computational resources for validation and since the missing values problem was not the focus, I randomly picked

[1] The dataset is available at `http://www.ncbi.nlm.nih.gov/geo` under accession number GSE7390.

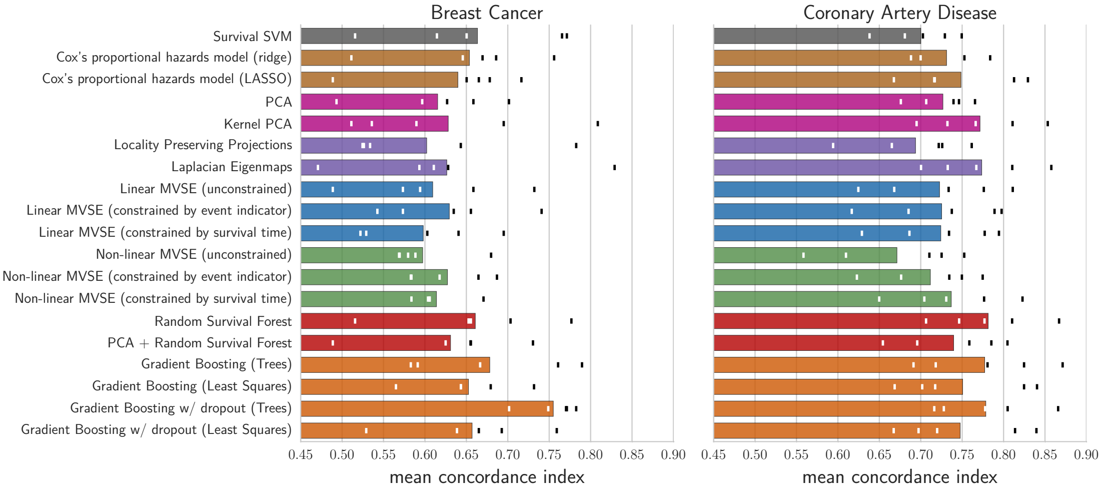


**Figure 6.1**: Harrell's concordance index of 5-fold cross-validation after hyper-parameter search. Horizontal bars indicate the average performance across all folds, and vertical markers the performance of individual folds. If not mentioned otherwise, a linear survival support vector machine was chosen as survival model for feature extraction methods.

one multiple imputed dataset, which was used for all subsequent analyses. Finally, continuous variables were normalized to have zero mean and unit variance.

### 6.3.2 Results

I will briefly summarize the results depicted in figs. 6.1 and 6.2, which serve as basis for the subsequent discussion concerning the questions posed earlier on page 139.

Survival models trained on the breast cancer dataset (fig. 6.1 left) generally performed modestly with mean $c$ index below 0.7 for all experiments, except gradient boosting with regression trees as base learners and using dropout. Interestingly, the results show that combining survival SVM with any feature extraction method resulted in a loss of predictive performance when compared to the baseline (without any dimensionality reduction). Moreover, I observed that linear singleview spectral embedding techniques (PCA, LPP) performed considerably worse than their non-linear counterparts (Kernel PCA, LE). For the remaining single- and multiview algorithms, the performance was comparable.

Results on the coronary artery disease dataset (fig. 6.1 right) looked quite different. Here, all feature extraction methods resulted in an improved mean $c$ index, except LPP and unconstrained non-linear MVSE. Kernel PCA and LE, both non-linear methods, performed equally well and better than their linear counterparts PCA and LPP, respectively. In the multiview setting, the choice of neighborhood graph construction made no considerable difference when using linear MVSE, in contrast to non-linear MVSE, where restricting links in the neighborhood graph to patients with similar survival time resulted in a considerably better performance than alternative methods (unconstrained, constrained by event indicator). Nevertheless, the performance difference between the best performing variants of linear and non-linear MVSE was minor, and both were outperformed by Kernel PCA and LE. Overall, non-linear survival models with embedded feature selection, namely random survival forest and gradient boosting with regression trees as base learners, performed best.

Experiments using the Framingham Offspring dataset revealed that all methods achieved a fair performance and that the performance differences among methods was smaller compared to results on the other two datasets (fig. 6.2 top left). In addition, the variance among cross-validation folds was considerably lower compared to the other experiments. Most notably, the performance of non-linear MVSE trailed the remaining methods. As for the coronary artery disease dataset, random survival forest and gradient boosting performed best.

### 6.3.3 Predictive Performance

Based on results presented in the previous section, I will now return to questions 1-3 on page 139 regarding the performance differences between single- and multiview spectral embedding as well as feature selection methods.

**Question 1.** Surprisingly, results indicate that ignoring the original source of features and simply assuming features originate from a common vector space does not decrease predictive performance, as is evident by the results of LE and LPP compared to MVSE. Therefore, I have to conclude that a multiview approach did not provide any benefits over singleview algorithms when considering the performance of survival models after dimensionality reduction.

This result is orthogonal to works presented in section 6.1.2, where experiments were carried out with respect to various classification problems in computer vision. I see three main differences to my experiments: 1) datasets in computer vision are usually much larger than in the medical domain, 2) the vast majority of image descriptors produce continuous valued feature, i.e., feature vectors are homogeneous, and 3) previous experiments were focusing on classification problems and encompassed performance measures other than the ones used here to judge the benefits of multiview spectral embedding algorithms. Considering these differences, it is not surprising that previous results do not apply when studying survival analysis.

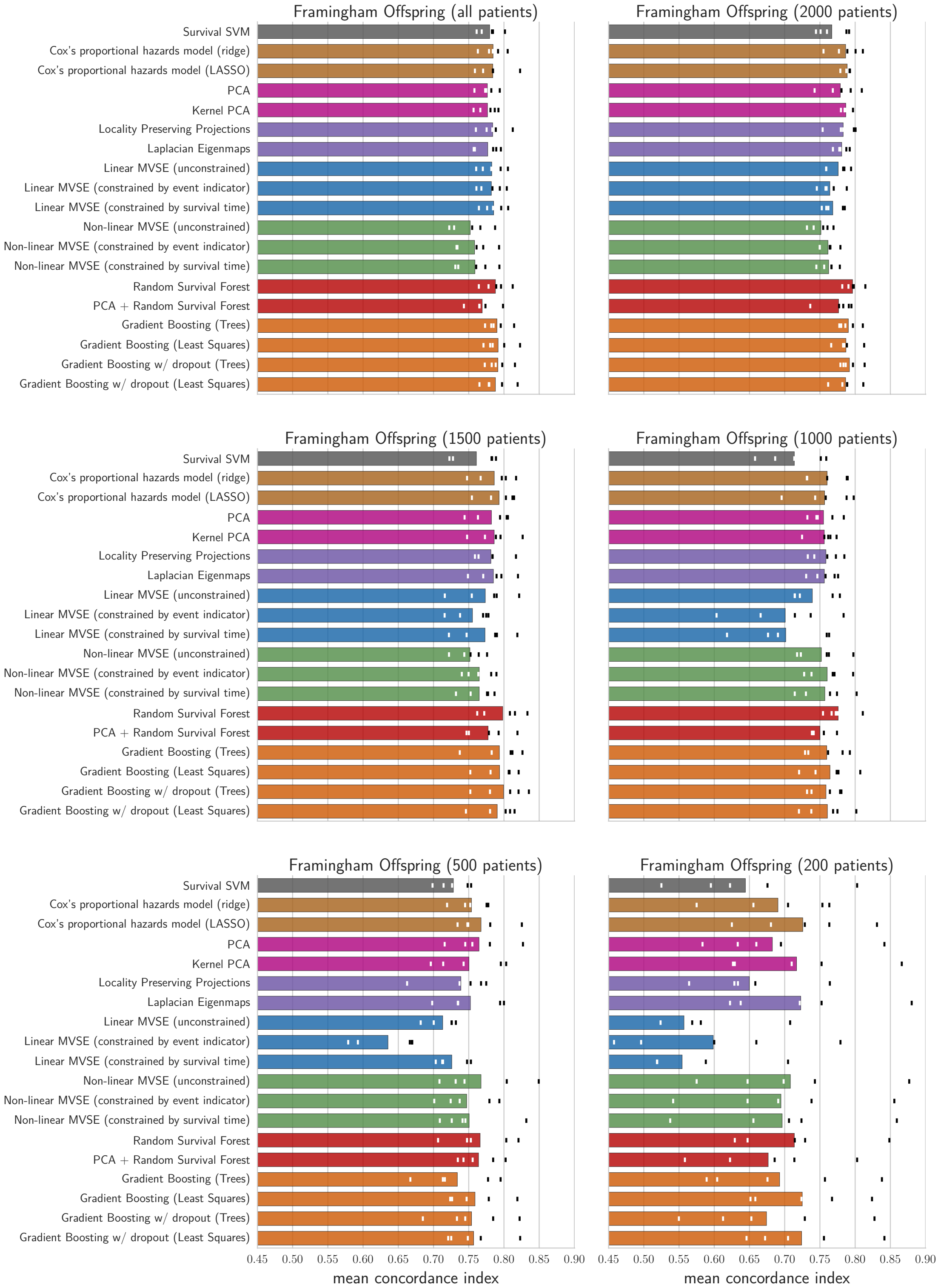


**Figure 6.2**: Harrell's concordance index of methods on random subsamples of varying size of the Framingham Offspring dataset.

**Question 2.** Turning to the second question, whether non-linear spectral embedding algorithms can improve the capabilities of a linear survival model. Results in figs. 6.1 and 6.2 do not convey a clear message with respect to this question. Whereas all feature extraction methods failed on the breast cancer dataset (fig. 6.1 left), they did improve the performance of linear survival SVM on the remaining two datasets (fig. 6.1 right, and fig. 6.2 top left). In my experiments, the breast cancer dataset was the one with the least number of samples, whereas the next bigger dataset was almost by an order of magnitude larger (1,233 patients for the coronary artery disease data). Therefore, a possible reason for the poor performance of spectral embedding algorithms could be the amount of data. Usually, if more data is available, the underlying manifold can be sampled more densely, which leads to a more accurate recovery. To determine whether feature extraction methods fail due to the number of samples in the data and not due to one dataset being more difficult than the other, I performed another set of experiments by randomly subsampling the Framingham Offspring dataset to 200, 500, 1000, 1500, and 2000 patients and recording the performance of all nineteen methods on each subsample. Figure 6.2 summarizes the results on all five subsets.

The results clearly indicate that dataset size plays a crucial role in the performance of methods. With only 200 patients, the performance differences between methods were most pronounced, ranging between mean $c$ index of 0.554 (linear MVSE) and 0.726 (Cox's proportional hazards model with LASSO penalty). In addition, results are characterized by a high variance between cross-validation folds for all methods. With increasing number of patients, the overall performance increased, and differences between methods as well as between cross-validation folds decreased. Consequently, most feature extraction methods can only succeed if a sufficient number of samples from the underlying manifold are available. In particular, I experienced that linear MVSE suffers from numerical instabilities and convergence problems with small sample sizes, which is evident from results with respect to the subsample with 200 patients (fig. 6.2 bottom right), where its performance is close to random guessing ($c$ index = 0.5). Problems related to recovering the smallest eigenvalues have been discussed by Van Der Maaten *et al.* [302] too.

In addition to problems caused by small sample size, the presence of noise or uneven sampling of the underlying manifold can be an issue for spectral embedding algorithms. In particular, feature extraction methods relying on local neighborhoods (LE, LPP, MVSE) are subject to these effects. If the underlying manifold is corrupted by noise, global methods are likely to give better results than local methods [245]. Alternatively, one could use empty region graphs instead of $k$ nearest neighbor graphs to define local neighborhoods [63]. To investigate whether results are affected by noise and whether the proposed definition of locality based on random survival forest is suitable, I compared the performance of local methods to the performance of global methods, namely PCA and Kernel PCA. Experiments revealed no systematic impairment of local methods and hence I concluded that my proposed neighborhood graph construction scheme properly captures locality of heterogeneous feature vectors.

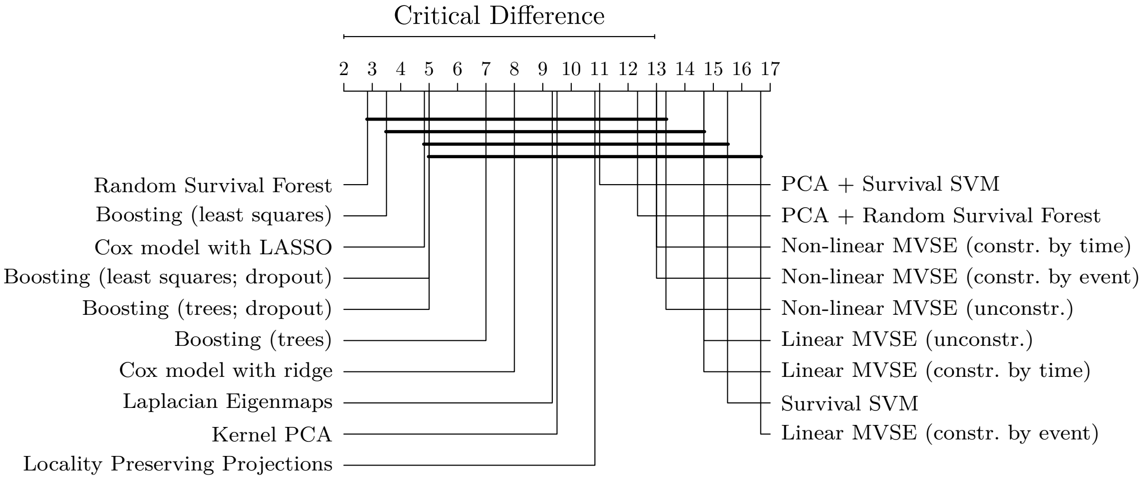


**Figure 6.3**: Comparison of methods with the Nemenyi post-hoc test [75] on the full and subsampled Framingham Offspring dataset. Methods are sorted by average rank (left to right) and groups of methods that are not significantly different ($p$-value $> 0.05$) are connected.

**Question 3.** Question 3 concerns the comparison of survival models trained on a low-dimensional representation of the training data to survival models with embedded feature selection. The experiments revealed that feature selection methods proved to be more stable than feature extraction methods with small samples sizes. Generally, the performance difference between feature selection methods was relatively small. Methods specifically designed to achieve sparsity in the number of features – Cox's proportional hazards model with LASSO penalty, gradient boosting with componentwise least squares, and random survival forest – excelled if the ratio between number of features and samples was close to one.

Moreover, I investigated whether one or more methods significantly outperformed the others, disregarding the size of the dataset. Hence, I combined the cross-validation results of all five subsampled datasets as well as the original Framingham Offspring data and determined whether any two survival models significantly differ from each other using Friedman's test and the Nemenyi post-hoc test [75]. Figure 6.3 shows that nine pairwise comparisons resulted in significant differences at significance level 0.05 (adjusted for multiple testing). First, linear MVSE constrained by event indicator performed significantly worse than 3 alternative methods: random survival forest, gradient boosting with shrinkage and componentwise least squares, and Cox's proportional hazards model with LASSO penalty. Moreover, random survival forest performed significantly better than the remaining variations of linear MVSE (unconstrained and constrained by survival time). Lastly, the linear survival support vector machine was outperformed by gradient boosting with shrinkage and componentwise least squares as

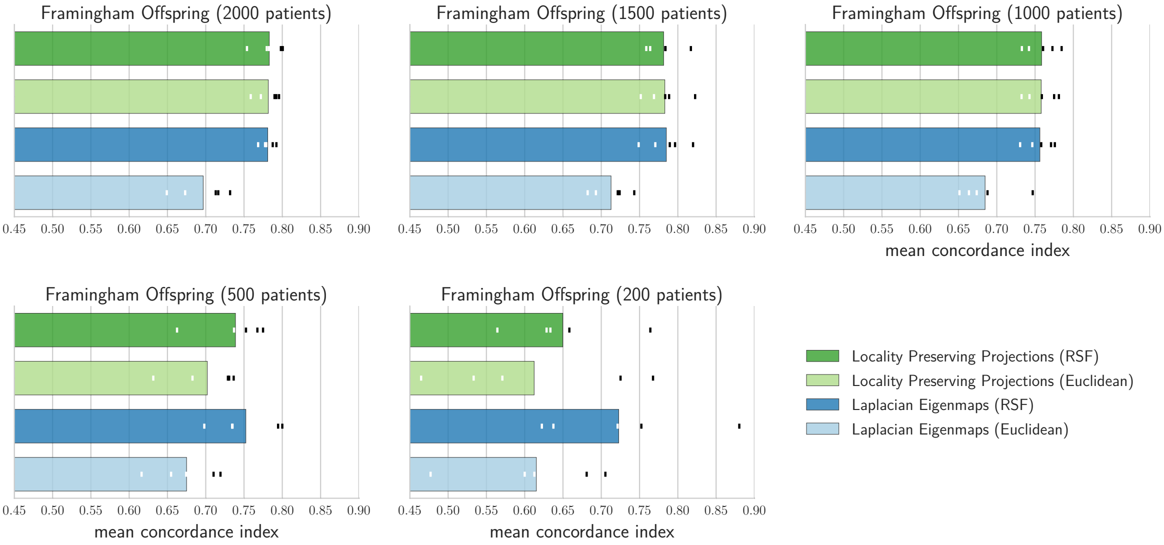


**Figure 6.4**: Comparison of survival models trained on low-dimensional embedding of Locality Preserving Projections and Laplacian Eigenmaps. Neighborhood graphs were either constructed according to the Euclidean distance among samples (and applying the heat kernel), or by Random Survival Forests (RSF) according to eq. (6.9).

well as random survival forest. From these results, I concluded that overall the most reliable choice are ensemble methods: random survival forest or gradient boosting with componentwise least squares as base learner.

### 6.3.4 Neighborhood Graph Construction

Now I will focus on question 4 concerning the construction of a neighborhood graph from heterogeneous survival data with random survival forests. In the last paragraph with respect to question 2 on page 147, I already mentioned that feature extraction methods operating on global and local scale performed comparably, which is evidence that local neighborhoods can be accurately captured by a random survival forest. Results indicate that constraining edges in the neighborhood graph slightly improved the concordance index if survival models were paired with non-linear MVSE, but not if paired with linear MVSE. However, experiments provided no clear indication whether constraining neighborhoods to patients with identical event indicator or similar survival time is preferred.

In addition, I investigated the performance of survival models if trained on a low-dimensional embedding of the training data, computed from the Euclidean distance between samples. Results in fig. 6.4 demonstrate that using random survival forests for constructing the neighborhood graph performs as good as the Euclidean distance and is the preferred choice when analyzing small sample size data. The difference between LE methods is larger compared to LPP methods, because LE uses the same distance measure during training and during prediction – to interpolate the location of a new sample on the manifold. If an improper distance measure is used, its error will be amplified, which is evident from results of LE in fig. 6.4.

Finally, I proposed using random survival forests to address the out-of-sample problem of non-linear spectral embedding methods. In my experiments, the performance of non-linear methods was overall superior to its linear counterparts, i.e., LE outperformed LPP and non-linear MVSE outperformed linear MVSE, which strengthens my observations from the comparison of LE with RSF-based distance to LE with Euclidean distance. Hence, I concluded that the proposed interpolation scheme accurately determines the location of new data points on the existing manifold of the training data.

### 6.3.5 The Utility of Views' Coefficients to Identify Clinical Markers

Next, I will turn to question 5 that studies whether view-specific weights $\boldsymbol{\alpha}$ in the MVSE algorithm provide insight into which group of features (view) plays the most important role in a particular disease. For clinical purposes, the predictive performance is not the only factor in determining whether a method is suitable or not, as outlined in the beginning of section 6.2 on page 132. Often the primary objective is to identify factors and biological pathways that are most decisive in diagnosing a disease. Therefore, methods should be able to robustly identify the most important group of features.

As mentioned in section 6.2.2, setting the complementary factor $r$ to a value close to 1 results in assigning more weight to a single view when solving eq. (6.5). Therefore, I performed a set of experiments to investigate whether view-specific weights $\alpha_k$ can be used to select clinically meaningful views. Experiments consisted of repeatedly learning a low-dimensional embedding on 50 subsamples of size equal to 50% of datasets' original size (drawn without replacement) and recording coefficients $\boldsymbol{\alpha}$. The complementary factor $r$ was gradually lowered from 5.0 to 1.1 in steps of 0.1, while keeping the remaining hyper-parameters fixed (number of nearest neighbors, dimensionality of embedding).

Figure 6.5 demonstrates the results on the coronary artery disease data[2] and shows the effect of the complementary factor $r$ as it gets closer to 1. With $r$ being large, weights $\boldsymbol{\alpha}$ are assigned roughly equally to all views and as $r \to 1$, more weight is assigned to a single view. Furthermore, linear and non-linear MVSE associated the highest weight to different views. Linear MVSE picked the angiography view as the most important view, disregarding constraints on the neighborhood graph. In contrast, the choice of

[2] Results with respect to the Framingham Offspring data can be found in appendix A.3 on page 196.

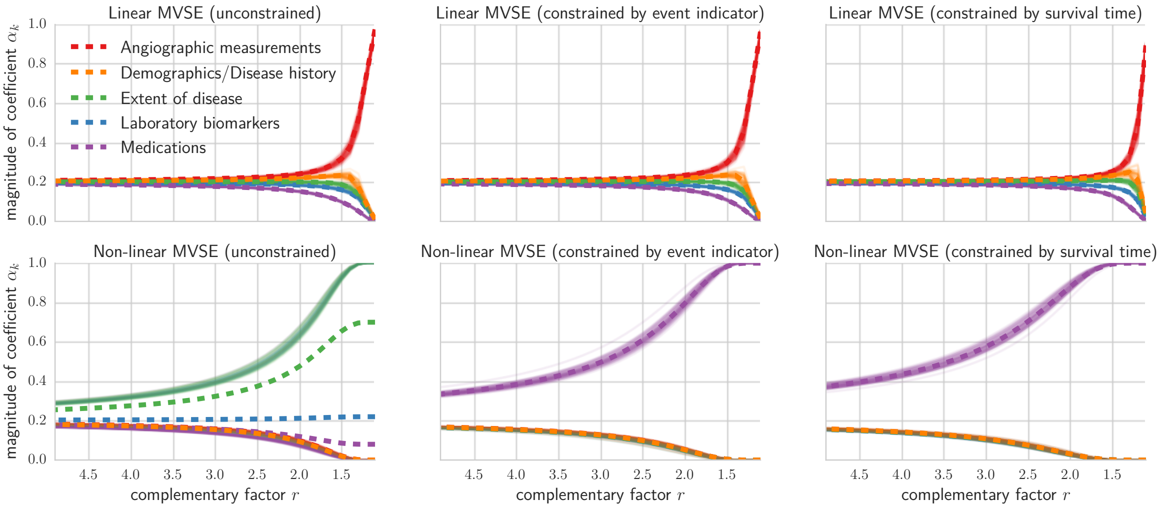


**Figure 6.5**: Value of view-specific coefficients for varying complementary factor. The complementary factor $r$ in multiview spectral embedding (MVSE) was chosen to be $r \in \{1.1, 1.2, \ldots, 4.9, 5\}$. Solid lines indicate the path of view-specific coefficients in 50 random subsamples of the coronary artery disease data. The average coefficient across all subsamples is indicated by dashed lines.

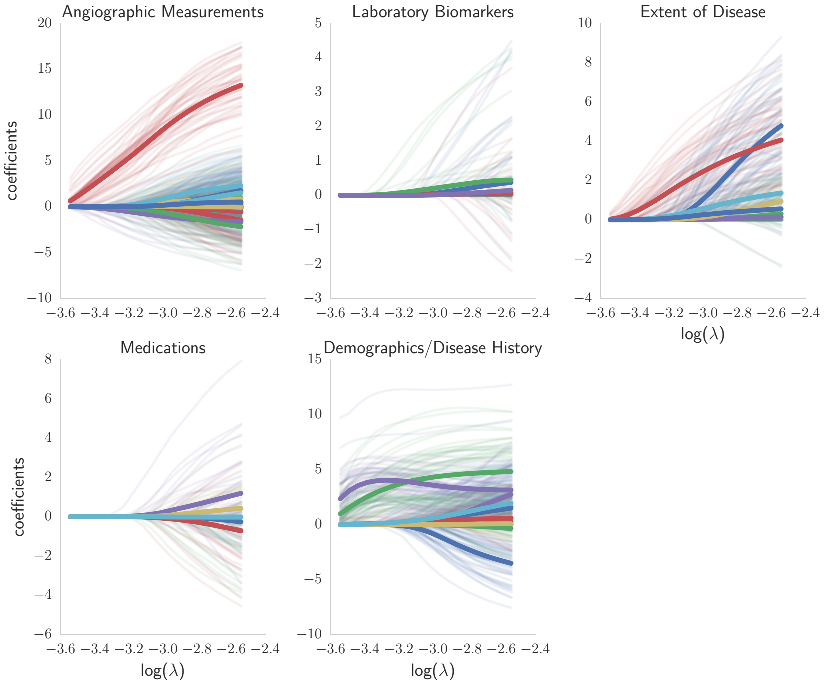


**Figure 6.6**: Coefficients of Cox's proportional hazards model with group LASSO penalty. Solid lines indicate the path of coefficients across 20 fixed values of the regularization parameter $\lambda$ from 50 random subsamples of the coronary artery disease data. The average coefficient across all subsamples is indicated by bold lines.

constraint made a substantial difference with non-linear MVSE. Medication was the most important view when constraining connections in the neighborhood graph, but results differed dramatically for unconstrained non-linear MVSE; it showed a high degree of instability in assigning one view the highest importance across subsamples – alternating between the extent of disease, biomarkers and medications. A possible explanation for the highly unstable results might be that local neighborhoods are badly preserved among multiple subsamples of the data. Thus, when repeatedly subsampling the data, the local neighborhood of samples might have changed and led to a different embedding and ultimately to different view-specific coefficients. Constraining connections in the neighborhood graph seemed to improve the overall stability of non-linear MVSE.

Furthermore, I compared the importance of views determined by MVSE to those determined by Cox's proportional hazards model with group LASSO penalty (see [327] and section 3.2.5 on page 43). The group LASSO behaves similar to the regular LASSO penalty, but couples the coefficients of a group of variables: either all variables of a group enter the model (non-zero coefficients), or all variables are excluded (zero coefficients). Thus, I grouped features according to their associated view and ran the same experiment as above. Figure 6.6 depicts the paths of coefficients for all features, separated by view. In the predominant number of runs ($> 92\%$), the demographics and disease history, and angiographic measurements view got "activated" first, and medications last, which is in accordance with results obtained from linear MVSE in fig. 6.5. In addition, current treatment guidelines for coronary artery disease (e.g. [318]) stress the importance of angiographic measurements in determining optimal treatment. Therefore, I believe that view-specific weights of linear MVSE can potentially provide valuable insight into the most important factors of a disease of interest. With respect to non-linear MVSE, I am not aware of any clinical study that would support the finding that medications outweigh all other factors, which led me to conclude that results of non-linear MVSE are ultimately implausible.

### 6.3.6 Conclusion

At the beginning of section 6.3 (page 139), I posed five questions concerning single- and multiview spectral embedding methods. I investigated these questions by implementing a comprehensive empirical study of 19 feature selection and feature extraction algorithms for building survival models from heterogeneous medical records. From these experiments I arrived at the following conclusions:

1. the performance difference between survival models paired with singleview spectral embedding algorithms and those paired with multiview spectral embedding algorithms was minor when using random survival forests to construct neighborhood graphs,

2. feature extraction methods are a valuable alternative to feature selection methods if the main interest is in improving predictive performance and if the dataset is large enough such that the number of samples is sufficient to describe the underlying manifold – in the experiments presented here, using less than 500 patients led to poor results,

3. embedded feature selection is the preferred choice for datasets with small sample size, because, in this situation, spectral embedding algorithms suffer from an insufficient number of samples from the underlying manifold,

4. random survival forests can be used to address two of the main problems encountered with methods based on spectral embedding: neighborhood graph construction in the presence of heterogeneous feature vectors and out-of-sample extension for new data points,

5. linear MVSE provides insight into the relevance of views similar to Cox's proportional hazards model with group LASSO penalty, which could potentially help to understand a view's role in a particular disease.

# 7 Predicting Survival of Prostate Cancer Patients

## 7.1 Prostate Cancer DREAM Challenge

### 7.1.1 Overview

Prostate cancer is currently associated with the highest incidence rate of all cancer types among men in Europe [95] and the United States of America [265]. It is estimated that prostate cancer accounts for 22.8% of new cancer cases in Europe and 26% of new cases in the U.S. each year [95, 265]. Among all cancer deaths, an estimated 9-10% are due to prostate cancer each year, which ranks the mortality rate of prostate cancer second after lung cancer in the U.S. [265], and third after lung cancer and colon/rectum cancer in Europe [95].

Depending on the stage of prostate cancer, treatments include surgery, radiation therapy, chemotherapy, and hormone therapy. The latter prevents the cancer from growing by lowering levels of androgens, which are a class of steroid hormones (e.g. testosterone). For instance, drugs blocking androgen receptors or the production of androgens in the andrenal glands reduce androgens to castrate levels and thus limit cancer growth. Although androgen deprivation therapy (ADT) is often successful in the early stages of therapy, in 10-20% of prostate cancer patients the cancer will inevitably progress from castrate-sensitive to castrate-resistant within 5 years [174]. The median survival time of individuals with castrate-resistant prostate cancer (CRPC) is typically less than 2 years [62, 174]. In addition, bone metastasis are often an attendant symptom of CRPC, which further complicates treatment [174]. Today, a number of treatment regimes have been developed for patients with metastatic castration-resistant prostate cancer, including chemo- and radiation therapy (docetaxel, cabazitaxel and radium-223) and agents affecting the immune system (sipuleucel-T) or androgen pathways (abiraterone and enzalutamide) [62, 223]. Despite these advances, the impact on survival still remains modest – improvements are typically measured in months – and the optimal sequence of therapies or combinations thereof are mostly unknown [62, 321].

The aim of the Prostate Cancer DREAM Challenge [65] was to expose the research community to a large and curated set of patients with metastatic, castrate-resistant prostate cancer (mCRPC) to foster the development of new models that could ultimately provide a better understanding of factors affecting overall survival of patients. Participants had access to medical records of 1,600 mCRPC patients, including follow-up, and were tasked with two different subchallenges during a three month period. The first of two subchallenges focused on predicting overall survival, whereas the second subchallenge focused on predicting treatment discontinuation due to adverse events.

### 7.1.2 Challenge Questions

#### Prediction of Survival

The objective of the first subchallenge was to build a prognostic model to predict overall survival of mCRPC patients. The model proposed by Halabi *et al.* [121] was used a baseline. They used Cox's proportional hazards model with adaptive LASSO penalty (see [328] and section 3.2.5 on page 43) to select nine out of 22 variables. Six out of the nine variables corresponded to the concentration of biomarkers: lactate dehydrogenase, albumin, hemoglobin, alkaline phosphatase and prostate-specific antigen. Moreover, they included the Eastern Cooperative Oncology Group (ECOG) performance status, current use of opioid analgesics and the disease site (metastasis in lymph nodes only, in lymph nodes and bones, or any other visceral metastases). The model was constructed from a set of 1,050 mCRPC patients and evaluated on an independent set of 942 men from a separate trial. The final model achieved an integrated area under the time-dependent, cumulative-dynamic ROC curve of 0.76 on the hold-out data.

Models that were considered eligible for the first subchallenge had to demonstrate statistical improvement in predicting overall survival compared to the model by Halabi *et al.* [121]. Newly proposed models were evaluated based on two criteria: 1) how well predicted risk scores rank patients according to actual survival time, and 2) how well models predict the exact time of death. The first criteria was determined by the area under the time-dependent, cumulative-dynamic ROC curve (see section 3.7 on page 83). The second criteria was evaluated based on the root mean squared error (RMSE) with respect to deceased patients in the test data. Whereas the first criteria corresponds to a traditional evaluation measure for survival models, the second criteria evaluates the exact number of days till death and is more challenging.

Participants could build their model from any information that can be extracted from health records of 2,070 mCRPC patients. During the challenge, participants were offered the opportunity to evaluate and refine their models on a small test set of 157 subjects. Evaluation was performed automatically and each team could submit predictions of five models at three fixed dates during the challenge period. Hence, participants could use the feedback to improve their methods for subchallenge one while the challenge was ongoing.

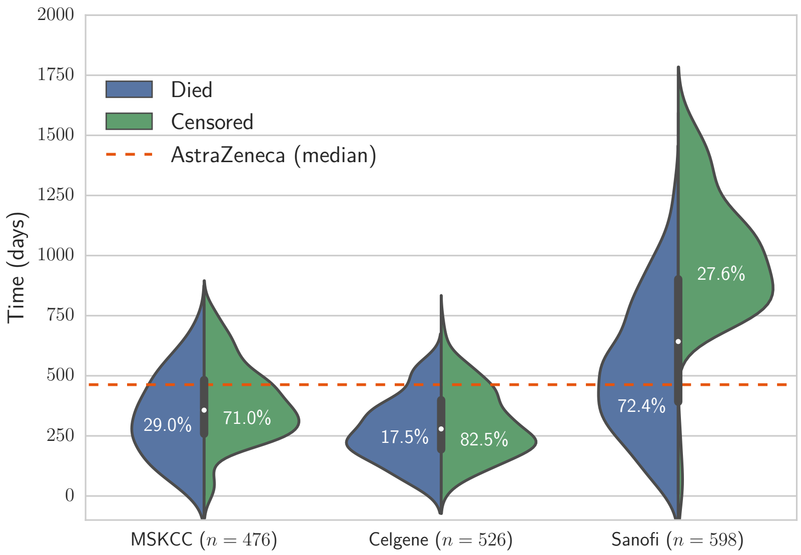


**Figure 7.1**: Overview of distribution of survival and censoring times in training data from Memorial Sloan Kettering Cancer Center (MSKCC), Celgene and Sanofi. Numbers in brackets denote the total number of patients in the respective study, and the dashed line is the median follow-up time in the AstraZeneca study, which was used as independent test data.

### Prediction of Treatment Discontinuation

The intent of the second subchallenge was to propose models that can predict whether mCRPC patients treated with docetaxel – an agent used in chemotherapy – are going to experience adverse events within three months after treatment began. Adverse events can arise from side effects of chemotherapy, which may require discontinuation of treatment if symptoms become life-threatening. Therefore, knowing the risk of severe complications caused by chemotherapy would allow considering alternative treatment options early. However, factors that lead to discontinuation of treatment are still unknown and no published work proposed a model to predict discontinuation as of the start of the Prostate Cancer DREAM Challenge [65]. Thus, no baseline model was available for this subchallenge and models were solely compared based on the area under precision-recall curve (AUCPR). Moreover, no intermediate evaluation on the test data until the final submission was possible.

## 7.1.3 Data Description

Participants were provided access to patients' health records from three separate phase III clinical trials [34, 223, 283] for training, and data from an independent, unpublished clinical trial of 470 men for testing (values of dependent variables were held back and not revealed to participants). An overview is depicted in fig. 7.1.

Patient's health records comprised a wide range of clinical information: prior medications, comorbidities, tumor measurements, laboratory tests, and vital signs (see fig. 7.2). Most data was collected before the first treatment, except for tumor and lab measurements, where data was collected after treatment began too. In all cases, dates were provided as the number of days relative to the first treatment date, which means negative values indicate a pre-treatment measurement. Due to the nature of the overall data being collected from four individual clinical studies, there were considerable differences among the four trials, which had to be addressed when constructing models. For instance, there was a large disparity in the distribution of censoring and survival times (see fig. 7.1). Moreover, not all studies recorded data to the same extent and level of detail, which resulted in a high amount of missing values for some features as well as inconsistencies in nomenclature.

In particular data contributed by the Memorial Sloan Kettering Cancer Center (MSKCC) did not contain detailed information about a patient's tumors and comorbidities (marked orange in fig. 7.2). For patients of this study only summary information for a fixed set of tumor locations and a fixed list of medications were accessible. Moreover, several attributes were not recorded in the AstraZeneca study, which forms the test data, and consequently these attributes could not be used for the final model (marked red in fig. 7.2). Table 7.1 provides a rough overview to which extent and level of detail data was available from the four studies. Note that the numbers do not reflect whether terms were consistent across all studies (e.g. for body systems).

To build a model for any of the subchallenges, data had to be aggregated on a per patient basis, inconsistencies had to be resolved, and a set of features had to be composed that can be extracted from raw training data as well as raw test data. Next, I will provide a detailed description on how these issues were addressed.

## 7.2 Extracting Features Describing Patient Records

### 7.2.1 Patient-level Aggregation

In a first step, I aggregated information stored in tables containing data about prior medications, comorbidities, tumor measurements, laboratory tests, and vital signs. To obtain a common set of features for all patients, I extracted features describing medications, comorbidities, and so forth, based on how often they occurred in the respective tables across all studies. This involved manually resolving inconsistencies of terms used in different studies, because often there were subtle differences such as

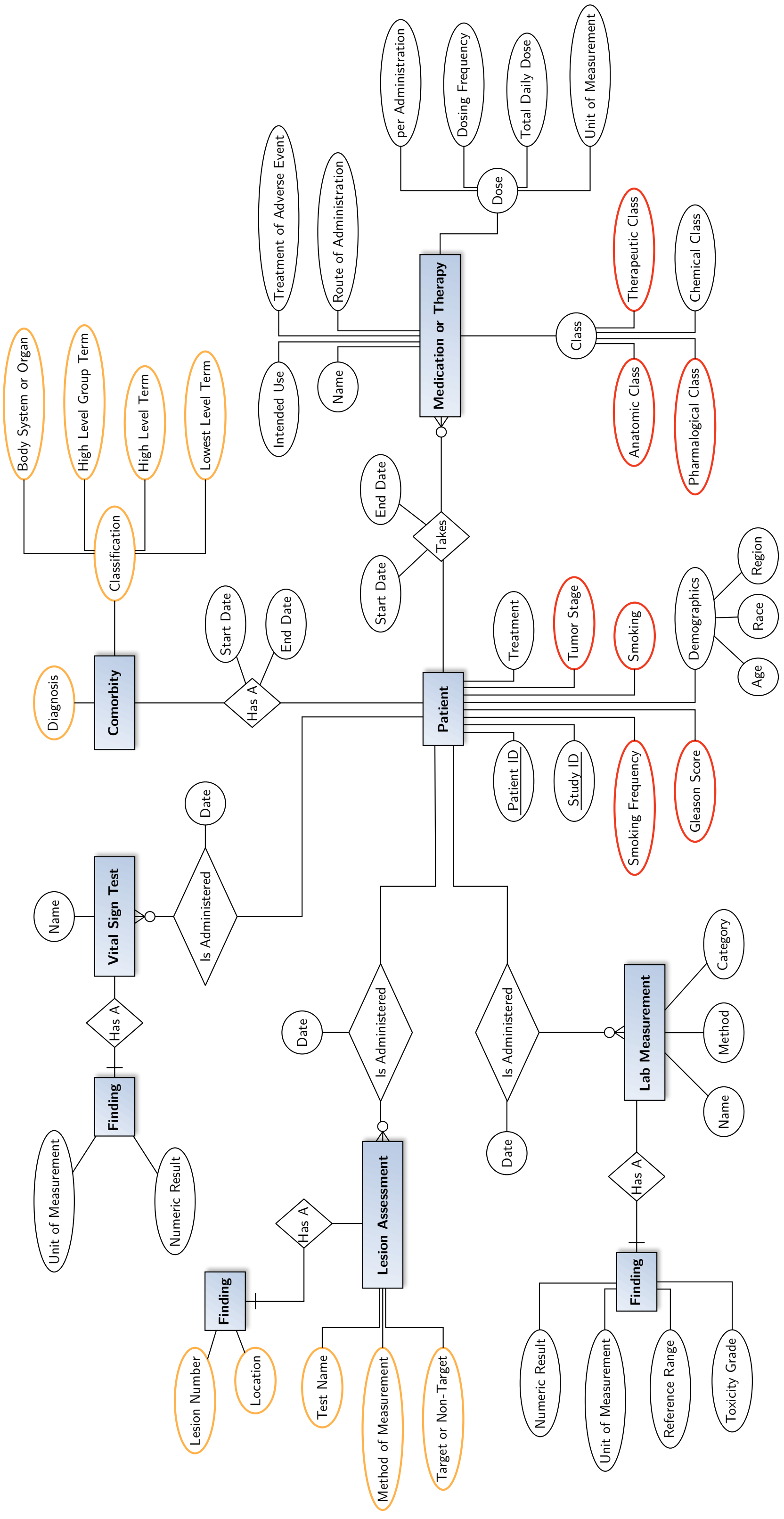


**Figure 7.2**: Entity-relationship diagram summarizing the structure of data in the Prostate Cancer DREAM challenge. Attributes marked red were not available in the AstraZeneca trial, which formed the test data, and attributes marked orange were not available in the study by the Memorial Sloan Kettering Cancer Center.

**Table 7.1**: Characteristics of data from four studies used in the Prostate Cancer DREAM challenge. Values marked by an asterisk were only available in aggregated form.

| | MSKCC | Celgene | Sanofi | AstraZeneca |
|---|---|---|---|---|
| Age | | | | |
| 18 – 64 | 111 | 171 | 219 | 160 |
| 65 – 74 | 211 | 246 | 254 | 217 |
| $\geq 75$ | 154 | 109 | 125 | 93 |
| # of Comorbidities | 13* | 743 | 683 | 572 |
| # of Body Systems/Organs | 24* | 25 | 25 | 25 |
| # of High level group terms | n.a. | 194 | 183 | 168 |
| # of High level terms | n.a. | 425 | 399 | 343 |
| # of Lowest level terms | n.a. | 1,084 | 987 | 883 |
| # of Lab tests | 13 | 65 | 67 | 51 |
| # of Medications/Therapies | 585 | 646 | 685 | 488 |
| # of Intended use | 1,056 | 1,220 | 3 | 3 |
| # of Chemical classes | 241 | 230 | 318 | 232 |
| # of Tumor tests | n.a. | 3 | 3 | 2 |
| # of Tumor locations | 19* | 233 | 24 | 18 |
| # of Vital measurements | 2 | 7 | 6 | 3 |

abbreviations used in one study and the full name in other studies (e.g. “antiinfectives” versus “antiinfect.”). Furthermore, some studies used a more detailed description of terms whereas others only contained higher level information, which was most prominent for medications (e.g. “antibiotics” versus “penicillins,” “cephalosporins” and “other antibiotics”).

After computing features based on the most common terms for medications, comorbidities, and so forth, the set of features had to be refined such that features can be computed for patients of most studies, in particular AstraZeneca, which was used for testing. Data with respect to a particular laboratory test were often only available from one study, but absent in other studies, which meant the lab test could not be used for model building, because of the high amount of missing values when considering all studies. The relevant set of features for each individual study was determined by only considering features that satisfied the following criteria: 1) it was available for at least 70% of patients within that study, 2) it had more than one unique value, and 3) its variance was higher than 0.001 (if continuous), or it had at least two categories that occurred more than ten times in the study’s data (if categorical). Finally, taking the intersection between all or a subset of these sets led to a common set of features, which resulted in seven sets of features as indicated in table 7.2. Data used for imputation of missing values included additional features that were *not* available for the test data (AstraZeneca), which is why fewer features were available for training models than for

**Table 7.2**: Different sets of features that were constructed by considering the intersection between studies in the Prostate Cancer DREAM challenge. Features used during imputation can be absent in the test data (AstraZeneca), whereas features for testing must be present in training and test data. Complete cases refers to the relative amount of samples free of missing values before imputation.

| MSKCC | Celgene | Sanofi | Samples | Features (Imputation) | Features (Testing) | Complete Cases |
|---|---|---|---|---|---|---|
| ● | ● | ● | 1,600 | 227 | 217 | 93.9% |
|  | ● | ● | 1,124 | 360 | 345 | 77.0% |
| ● |  | ● | 1,074 | 230 | 220 | 92.1% |
| ● | ● |  | 1,002 | 231 | 221 | 92.7% |
|  |  | ● | 598 | 382 | 350 | 64.0% |
|  | ● |  | 526 | 415 | 383 | 57.0% |
| ● |  |  | 476 | 242 | 223 | 78.8% |

imputing missing values. Next, I will provide a more detailed description about features describing medications, comorbidities, tumor measurements, lesion measurements, and vital signs.

**Medications.** Chemical classes and routes of administration were manually grouped into coarser groups than in the original data, which resulted in a total of 326 chemical classes and 20 routes of administration, from which features describing the 88 most often used chemical classes and the 5 most often used routes of administration were extracted. Each binary feature indicated whether a patient was ever taking any medication of this particular chemical class or route of administration. For six medications/therapies – hormone therapy, opium alkaloids, gonadotropin-releasing hormones, glucocorticoids, bisphosphonates, and anti-androgens – information about the duration of treatment was available. The duration of hormone therapy, gonadotropin-releasing hormones and anti-androgens was measured in one year intervals, whereas duration of opium alkaloids and glucocorticoids was measured in monthly intervals. Moreover, I added one binary feature describing whether a patient ever received medication for treatment of adverse events and one continuous feature corresponding to the total number of chemical classes a patient received.

**Comorbidities.** After resolving inconsistencies in the terms used to describe diagnoses, a total of 1,010 terms remained, from which binary features denoting the presence of each of the 67 most common comorbidities were extracted. Each of the binary features was associated with one additional feature representing the time of first occurrence, relative to the reference day of the respective study. Finally, I explicitly added a feature whether a patient experienced cancer pain and for how long, and the total number of recorded comorbidities for each patient.

**Lab Measurements.** First, only measurements that were conducted prior to treatment were considered when selecting the most common tests, because these denote baseline levels before treatment began. From all 85 tests, the 55 most common tests were selected. Multiple real-valued measurements were aggregated by computing the mean, multiple categorical measurements by computing the mode. For 40 lab tests, definitions of lower and/or upper bounds of a normal range were available and an additional categorical feature was added indicating whether the aggregated mean measurement was below, within, or above the respective reference range. Similarly, the date of measurement was available for 53 out of all selected lab tests, which was used to create another 53 features corresponding to the date of the first measurement, relative to reference day.

**Tumor Measurements.** A suitable set of lesion locations was determined by only considering records pre-dating the start of treatment and selecting the 9 most often occurring locations out of a total of 211 locations. Aggregation was performed by creating a categorical feature indicating whether no lesion, one lesion, or two or more lesions were found in a particular location. Moreover, each location was associated with a feature corresponding to the earliest day of assessment. Finally, the total number of target lesions and non-target lesions as well as minimum and maximum lesion size were extracted for each patient.

**Vital Signs.** Measurements were first converted to SI units and features describing the finding and date of pre-treatment measurements were extracted for all 10 performed tests. Multiple measurements per patient were aggregated by taking the mean.

### 7.2.2 Post-processing and Imputation

After extracting an initial set of features, post-processing was applied to adjust the scale of features and to discard meaningless features. First, continuous features with high skewness were log-transformed to obtain values more closely resembling a normal distribution. Transformation was automated by systematically computing the skewness of the empirical distribution and applying a log transformation if the skewness exceeded a threshold $\tau$; I empirically chose $\tau = 1.4$. Similarly, I applied an Anscombe transformation [9] to features corresponding to counts such that transformed values resembled a normal distribution. To reduce the impact of small changes in features corresponding to dates, I discretized those features by assigning each value a number according to which quartile the value belonged to. In the last step, useless features were discarded. A feature was considered useless if at least one of the following conditions applied:

- it was missing for more than 30% of samples,
- it comprised only one unique value (excluding missing values),
- its variance was smaller than 0.001 (if continuous), or it contained only a single category that occurred more than ten times in the training data (if categorical).

The number of remaining features after post-processing are summarized in column "Feature (Imputation)" in table 7.2.

### Imputation

Although most missing values could be eliminated by the relatively strict requirement that all features are present for at least 70% of patients of each individual study, between 6.1% and 43% of samples still contained one or more missing value (see the right most column in table 7.2). Hence, imputation of missing values was required before model-building.

To provide as much information as possible to the imputation process, I included features that were not available for the test data, but for participants of one or more other studies. Since data contained several hundred features, I opted for using a random survival forest for imputation (see section 3.6.3 on page 82) rather than multivariate imputation using chained equations, which would require considerably more computational time.

Study-specific datasets were imputed first, because they had the most amount of features available, followed by datasets consisting of subjects from multiple studies. For imputation of the challenge's test data (AstraZeneca), random survival forests were used as well, except that features that were not available in the test data (excluding dependent variables) were discarded from the training data (MSKCC, Celgene, Sanofi) before training a random survival forest on top of it. Subsequently, the resulting forest could be used to impute the test data.

Finally, all features that were absent in the test data were removed from imputed datasets listed in table 7.2 (see column "Features (Testing)") and the resulting training and test data – free of missing values – was used for all subsequent steps in my analyses (see appendix B.1 for a list of all features and their availability).

## 7.3 Considerations Regarding Prostate Cancer DREAM Challenge

I will now focus on the first subchallenge, where the objective was to develop models predicting survival of mCRPC patients.

After extracting features from data of all four studies, a survival model has to be trained on data from MSKCC, Celgene and Sanofi to predict survival of patients in the AstraZeneca study. However, due to the training data consisting of subjects from multiple clinical trials, several problems arise: 1) the median follow-up time differs drastically among studies and consequently the amount of censoring as well (see fig. 7.1), and 2) there are large differences in the set of features when considering data from a single study or combinations of multiple studies (see table 7.2).

The latter presumably has the biggest impact on a model's performance. If a model is trained on data from all available individuals, the sample size is maximized, but the number of features is restricted to the smallest common denominator of all studies. In contrast, when only taking data from a single study, a model can leverage all available features of that study, but is limited by a rather small sample size. The first approach might achieve higher performance on the test data because of a larger sample size, but also might miss out on important indicators. The second approach is characterized by a higher risk of overfitting due to small sample size, but including additional features might also increase the discriminatory performance. In addition to using data from all or only a single study, it could be beneficial to consider a combination of two studies, too. Even before choosing a particular survival model, one is confronted with the question "which of these approaches is preferred to train a survival model?"

To address these issues, it clearly is indispensable to consider alternative approaches to survival analysis beyond traditional models, such as Cox's proportional hazards model [67]. Next, I will first formulate heterogeneous survival ensembles, which were used in the first subchallenge, and then re-visit the question stated above.

## 7.4 Heterogeneous Survival Ensembles

Ensemble models have been successfully applied in machine learning [37, 78, 126] and survival analysis [148, 159, 239]. An ensemble comprises several base learners, whose predictions are aggregated to form an overall prediction. Aggregating multiple base learners provides an improvement over the prediction of a single base learner if base learners' predictions are accurate and at the same time diverse [78, 126]. The first requirement states that a base learner has to be better than random guessing and the second requirement that predictions of any two base learners must be uncorrelated. For instance, random forests satisfy the second condition by training each tree on a bootstrap sample of the original training data and by randomizing the split criterion in each node. It is easy to see that aggregating highly correlated predictions would not lead to a better overall prediction.

The base learners in most ensemble methods correspond to the same model, e.g., survival trees in a random survival forest [159] or regression trees in gradient boosting [104]. Caruana *et al.* [51] proposed *heterogeneous ensembles*, where base learners are selected from a library of many different learning algorithms, such as support vector machines, decision trees, $k$ nearest neighbor classifiers and gradient boosting models. In particular, the library itself can contain other (homogeneous) ensemble models such that the overall model is an ensemble of ensembles. Following Caruana *et al.* [51], constructing a heterogeneous ensemble consists of four steps:

1. Initialize an empty ensemble.
2. Update the ensemble by adding a model from the library that maximizes the (extended) ensemble's performance on an independent validation (hillclimb) set.
3. Repeat step 2 until the desired size of the ensemble is reached or all models in the library have been added to the ensemble.
4. Prune ensemble by reducing it to the subset of base learners that together maximize the performance on a validation (hillclimb) set.

By populating the library with a wide range of algorithms, the requirement of having a diverse set of base learners is trivially satisfied, although each model can be trained on a separate bootstrap sample of the training data as well. The requirement that base learners should be accurate is addressed by the second step in the algorithm of Caruana *et al.* [51] described above. Finally, the goal of the pruning step is to avoid overfitting on the validation set and to discard base learners that are likely to provide little benefit to the overall ensemble.

I adapted heterogeneous ensembles to build a large ensemble from a wide range of survival models to predict survival of mCRPC patients in subchallenge one. The main advantage of this approach is that it is not necessary to rely on a single survival model and any assumptions or limitations that model may imply. To the best of my knowledge, this is the first work that uses heterogeneous ensembles for survival analysis.

### 7.4.1 Efficient Ensemble Selection

During construction of a heterogeneous ensemble, an arbitrary error or performance measure can be optimized by selecting models from the library that maximize that particular performance on a validation set. As a result, the training data needs to be split into two non-overlapping parts: one part used to train base learners from the library, and another part used as validation (hillclimb) set. In domains with small sample size, which is the case for data from the Prostate Cancer DREAM challenge, this approach is problematic. Caruana *et al.* [50] observed that if the validation set is small, the ensemble tends to overfit more easily, which becomes especially concerning when the library is large. It would seem that heterogeneous ensembles are not adequate in situations when training data is scarce. In the same paper, Caruana *et al.* [50, p. 3] proposed a solution that "embed[s] cross-validation within ensemble selection so that all of the training data can be used for the critical ensemble hillclimbing step." Instead of setting aside a separate validation set, they proposed to use *cross-validated models* to determine the performance of a model from the library.

> **Definition 7.1: Cross-Validated Model.** A cross-validated model is itself an ensemble of identical models, termed *siblings*, each trained on a different subset of the training data. It is constructed by splitting the training data into $K$ equally sized folds and training one identically parametrized model on data from each of the $K$ combinations of $K-1$ folds. Together, the resulting $K$ siblings form a cross-validated model.

To estimate the performance of a cross-validated model, the complete training data can be used, because the prediction of a sample in the training data only comes from the sibling that did not see the particular sample during training. Therefore, the estimated performance based on all samples in the training data has the same properties as if one used a separate validation set, but without reducing the size of the training data. If a truly new data point is to be predicted, the prediction of a cross-validated model is the average of the predictions of its siblings. Algorithm 7.1 summarizes steps to build a heterogeneous ensemble from cross-validated survival models.

Note that if a cross-validated survival model is added to the ensemble, the ensemble actually grows by $K$ identically parametrized models of the same type – the siblings (see line 13 in algorithm 7.1). Therefore, the prediction of an ensemble consisting of $S$ cross-validated models is in fact the average of $K \times S$ models.

### 7.4.2 Ensemble Pruning

The ensemble selection algorithm 7.1 described in the previous section only ensures that base learners are accurate, but does not guarantee that predictions of base learners are diverse, which is the second important requirement for ensemble methods [78, 126]. One of the earliest approaches was proposed by Margineant and Dietterich [209]. They considered an ensemble of classifiers and used Cohen's kappa [60] to estimate the degree of disagreement of any pair of classifiers from the library on a validation set. Pruning was achieved by sorting all pairs of base learners according to the kappa statistic and picking $S$ pairs with the lowest kappa statistic.

Here, predictions are real-valued, because they either correspond to a risk score or to the time of death. Therefore, I adapted a method for pruning an ensemble of regression

**Algorithm 7.1:** Ensemble Selection for Survival Analysis.

**Input**: Library of $N$ base survival models, training data $\mathcal{D}$, number of folds $K$, minimum desired performance $c_{\text{min}}$.

**Output**: Ensemble of base survival models exceeding minimum performance.

```
$\mathcal{M} \leftarrow \varnothing$
for $i \leftarrow 1$ to $N$ do
    $\mathcal{C}_i \leftarrow \varnothing$
    for $k \leftarrow 1$ to $K$ do
        $\mathcal{D}^k_{\text{train}} \leftarrow$ $k$-th training set
        $\mathcal{D}^k_{\text{test}} \leftarrow$ $k$-th test set
        $M_{ik} \leftarrow$ Train $k$-th sibling of $i$-th survival model on $\mathcal{D}^k_{\text{train}}$
        $c_k \leftarrow$ Prediction of survival model $M_{ik}$ on $\mathcal{D}^k_{\text{test}}$
        $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{(\mathcal{D}^k_{\text{test}}, c_k)\}$   /* Store prediction and associated ground truth */
    end
    $\bar{c}_i \leftarrow$ Performance of $i$-th survival model based on predictions and ground truth in $\mathcal{C}_i$
    if $\bar{c}_i \geq c_{\text{min}}$ then
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{(M_{i1}, \ldots, M_{iK}, \bar{c}_i)\}$   /* Store $K$ siblings and performance of
        $i$-th model */
    end
end
return Base models in $\mathcal{M}$
```

models that accounts for a base learner's accuracy and its correlation to other base learners [241]. I will first describe the method by Rooney *et al.* [241] when the objective is to construct an ensemble of regression models (e.g. to predict the exact time of death) and subsequently propose a modification that enables pruning ensembles of survival models.

**Pruning Regression Ensembles.** Given a library of base learner, first, the performance of each base learner is estimated either from a separate validation set or via algorithm 7.1. To estimate a measure of diversity for a pair of regression models, Rooney *et al.* [241] utilized a model's residuals as a per-sample error measurement. Given the residuals of two models on the same test data, it is straightforward to obtain a measure of diversity by computing Pearson's correlation coefficient. They defined the diversity of a single model based on the correlation of its residuals to the residuals of all other models in the ensemble and by counting how many correlation coefficients exceeded a user-supplied threshold $\tau_{\text{corr}}$. The diversity score is then given by subtracting the number of correlated models from the total number of models in the ensemble and normalizing by the ensemble's size. If a model is sufficiently correlated with all other models, its diversity is zero, and if it is completely uncorrelated, its diversity is one. Finally, Rooney *et al.* added the diversity score of each model to its

**Algorithm 7.2:** Ensemble Pruning Algorithm of Rooney *et al.* [241].

```
Input: Set of base survival models $\mathcal{M}$ and their average cross-validation performance,
       validation data $\mathcal{D}_{\text{val}}$, desired size $S$ of ensemble, correlation threshold $\tau_{\text{corr}}$.
Output: Aggregated predictions of $S$ base survival models.
$c_{\max} \leftarrow$ Highest performance score of any model in $\mathcal{M}$
if $|\mathcal{M}| > S$ then
    $\mathcal{C} \leftarrow \varnothing$
    for $i \leftarrow 1$ to $|\mathcal{M}|$ do
        $p_i \leftarrow$ Prediction of data $\mathcal{D}_{\text{val}}$ using $i$-th base survival model in $\mathcal{M}$
        $count \leftarrow 0$
        for $j \leftarrow 1$ to $|\mathcal{M}|$ do
            $p_j \leftarrow$ Prediction of data $\mathcal{D}_{\text{val}}$ using $j$-th base survival model in $\mathcal{M}$
            if $i \neq j \wedge \text{correlation}(p_i, p_j, \mathcal{D}_{\text{val}}) \geq \tau_{\text{corr}}$ then
                $count \leftarrow count + 1$
            end
        end
        $d_i \leftarrow (|\mathcal{M}| - count)/|\mathcal{M}|$
        $\bar{c}_i \leftarrow$ Average cross-validation performance of $i$-th survival model in $\mathcal{M}$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{(i, \bar{c}_i/c_{\max} + d_i)\}$
    end
    $\mathcal{M}^* \leftarrow$ Top $S$ survival models with highest score according to $\mathcal{C}$
else
    $\mathcal{M}^* \leftarrow \mathcal{M}$
end
return Prediction of $\mathcal{D}_{\text{val}}$ by aggregating predictions of base learners in survival ensemble
    $\mathcal{M}^*$
```

accuracy score (measured relative to the best performing model in the ensemble) and selected the top $S$ base learners according to the combined accuracy-diversity score. A generalized version of the algorithm by Rooney *et al.* [241] is depicted in algorithm 7.2, where the correlation function would compute Pearson's correlation coefficient between residuals of the $i$-th and $j$-th model.

**Pruning Survival Ensembles.** The main difference of an ensemble of survival models to an ensemble of regression models is that a per-sample error measurement, similar to residuals in regression, generally does not exist. Instead, the prediction of a survival model consists of a risk score of arbitrary scale and the ground truth of the time of an event or the time of censoring. Obviously, a direct comparison of these values, e.g., by computing the squared error, is not meaningful. Therefore, I propose two alternative approaches: 1) computing Pearson's correlation coefficient between predicted risk scores, and 2) computing Kendall's rank correlation coefficient (Kendall's $\tau$) between predicted risk scores.

In contrast to the correlation between residuals, both proposed correlation measures for survival models are directly based on predicted risk scores and do not involve any ground truth. I believe this is not a disadvantage, because the combined score in line 15 of algorithm 7.2 already accounts for a model's accuracy, which could be estimated by the concordance index or integrated area under the time-dependent ROC curve on a validation set or using algorithm 7.1. In fact, since the diversity score for survival models does not depend on ground truth, the pruning step can be postponed until the prediction phase, under the assumption that prediction is always performed for a set of samples and not a single sample. Consequently, the ensemble will not be static anymore and is allowed to change if new test data is provided, resulting in a dynamic ensemble.

In summary, for pruning an ensemble of survival models, algorithm 7.2 is applied during prediction with the following modifications:

1. validation data $\mathcal{D}_{\text{val}}$ is replaced by feature vectors of the test data $\boldsymbol{X}_{\text{new}}$,
2. the performance score is based on the concordance index or integrated area under the time-dependent, cumulative-dynamic ROC curve (see section 3.7 on page 83),
3. the correlation is computed from predicted risk scores either using Pearson's correlation coefficient or Kendall's rank correlation coefficient.

The prediction of the final ensemble is the average predicted survival time of all its members after pruning.

*Note.* Harrell's concordance index (3.128) simplifies to Kendall's rank correlation coefficient in the absence of censoring.

## 7.5 Cross-Validation Results

I will now return to the question posed in section 7.3: Is it better to use all available features of a single study, or is it better to use all available samples from all studies? I want to emphasize that my results are entirely based on data from the Prostate Cancer DREAM challenge, as such, my conclusions are by no means general suggestions for arbitrary datasets.

### 7.5.1 Experiments

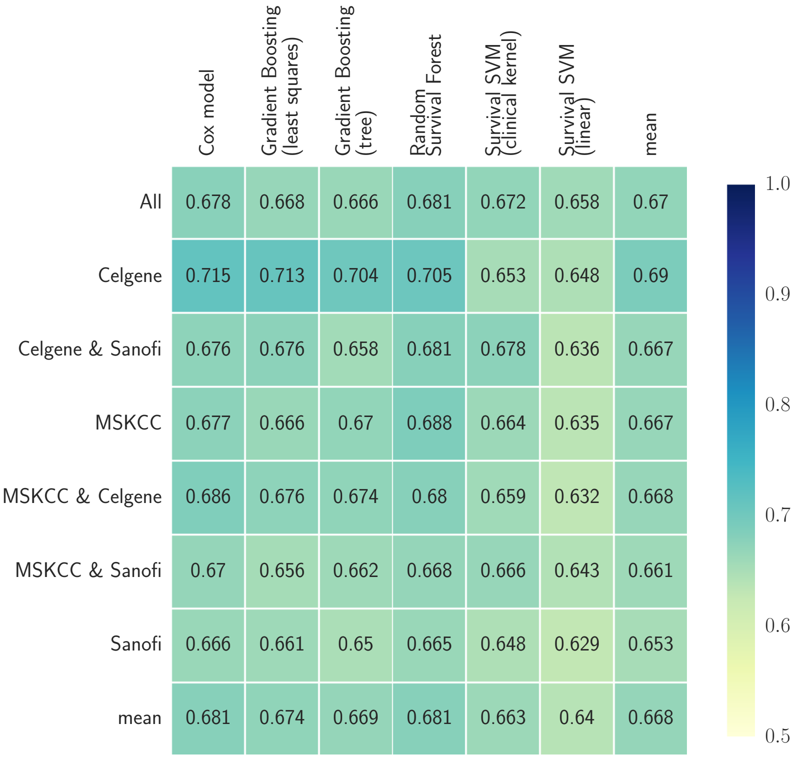
Since survival times of the AstraZeneca trial have not been revealed to participants, I will restrict myself to data from the remaining three studies for the following experiments. In the first experiment, I ran cross-validation on each of the possible combinations of datasets in table 7.2. Thus, test and training data contained individuals from the same combination of studies. In the second experiment, I used one of three datasets as hold-out data for testing and one or both of the remaining datasets for training. This setup resembles the challenge more closely, where test data corresponded to a separate study too. Each experiment was performed with each of the following six survival models:

1. Cox's proportional hazards model with ridge penalty (see [67] and section 3.2 on page 35),
2. Ranking-based Survival Support Vector Machine (see [236] and chapter 5 on page 99)
    a) as linear model without any kernel,
    b) or with clinical kernel (see [70] and section 3.3.5 on page 59),
3. (Stochastic) gradient boosting of negative log partial likelihood of Cox's proportional hazards model (see [239] and section 3.4 on page 60),
    a) with randomized regression trees as base learners [37, 39],
    b) or with componentwise least squares as base learners [45],
4. Random Survival Forest (see [159] and section 3.6 on page 79).

In addition, training of each survival model was wrapped by a grid search optimization to find optimal hyper-parameters. For each possible configuration of hyper-parameters, the complete training data was randomly split into 80% for training and 20% for testing to estimate a model's performance with respect to a particular hyper-parameter configuration. The process was repeated for ten different splits of the training data. Finally, a model was trained on the complete training data using the hyper-parameters that on average performed the best across all ten repetitions. Performance was estimated by Harrell's concordance index [127, 128], since it was used in evaluation of models submitted to the Prostate Cancer DREAM challenge too. All continuous features were normalized to zero mean and unit variance.

## 7.5.2 Results of Within-In Study Validation

Figure 7.3 summarizes the results when performing cross-validation for any of the seven datasets in table 7.2. Overall, the average concordance index ranged between 0.629 and 0.713 with a mean of 0.668. It is noteworthy that all classifiers but survival SVM models performed best on data of the Celgene study, which comprised 526 subjects and 383 features, which was the highest number of features among all studies. A survival SVM was likely at an disadvantage due to the high number of features and because feature selection is not embedded into its training as for the remaining models. In fact, survival SVM models performed worst on data from Celgene and Sanofi, which were the datasets with the most features. SVM-based models performed best if data from at least two studies were combined, which increased the number of samples and decreased the number of features. Moreover, the results show that linear survival support vector



| | Cox model | Gradient Boosting (least squares) | Gradient Boosting (tree) | Random Survival Forest | Survival SVM (clinical kernel) | Survival SVM (linear) | mean |
|---|---|---|---|---|---|---|---|
| All | 0.678 | 0.668 | 0.666 | 0.681 | 0.672 | 0.658 | 0.67 |
| Celgene | 0.715 | 0.713 | 0.704 | 0.705 | 0.653 | 0.648 | 0.69 |
| Celgene & Sanofi | 0.676 | 0.676 | 0.658 | 0.681 | 0.678 | 0.636 | 0.667 |
| MSKCC | 0.677 | 0.666 | 0.67 | 0.688 | 0.664 | 0.635 | 0.667 |
| MSKCC & Celgene | 0.686 | 0.676 | 0.674 | 0.68 | 0.659 | 0.632 | 0.668 |
| MSKCC & Sanofi | 0.67 | 0.656 | 0.662 | 0.668 | 0.666 | 0.643 | 0.661 |
| Sanofi | 0.666 | 0.661 | 0.65 | 0.665 | 0.648 | 0.629 | 0.653 |
| mean | 0.681 | 0.674 | 0.669 | 0.681 | 0.663 | 0.64 | 0.668 |

**Figure 7.3**: Cross-validation performance of survival models on data from Memorial Sloan Kettering Cancer Center (MSKCC), Celgene and Sanofi as well as any combination of these datasets. The last column (mean) denotes the average performance of all models on a particular data set and the last row (mean) denotes the average performance of a particular model across all datasets. Numbers indicate the average of Harrell's concordance index across five cross-validation folds.

machines performed poorly. A considerable improvement could be achieved when using kernel-based survival support vector machines with the clinical kernel, which is especially useful if data is a mix of continuous, categorical and ordinal features. For low-dimensional data, the kernel survival SVM could perform equally well or better than gradient boosting models, but always lacked behind random survival forest.

When considering the performance of models across all datasets (last row in fig. 7.3), random survival forest and Cox's proportional hazards model stood out with an average $c$ index of 0.681, outperforming the runner-up gradient boosting with componentwise least squares base learner. Random survival forest performed better than Cox's proportional hazards model on 4 out of 7 datasets and was tied on one dataset. The results seem to indicate that a few datasets contain non-linearities, which are modeled by random survival forest, but not by gradient boosting with componentwise least squares and Cox's proportional hazards model. Nevertheless, the latter performed as well as random survival forest when averaging results over all datasets.

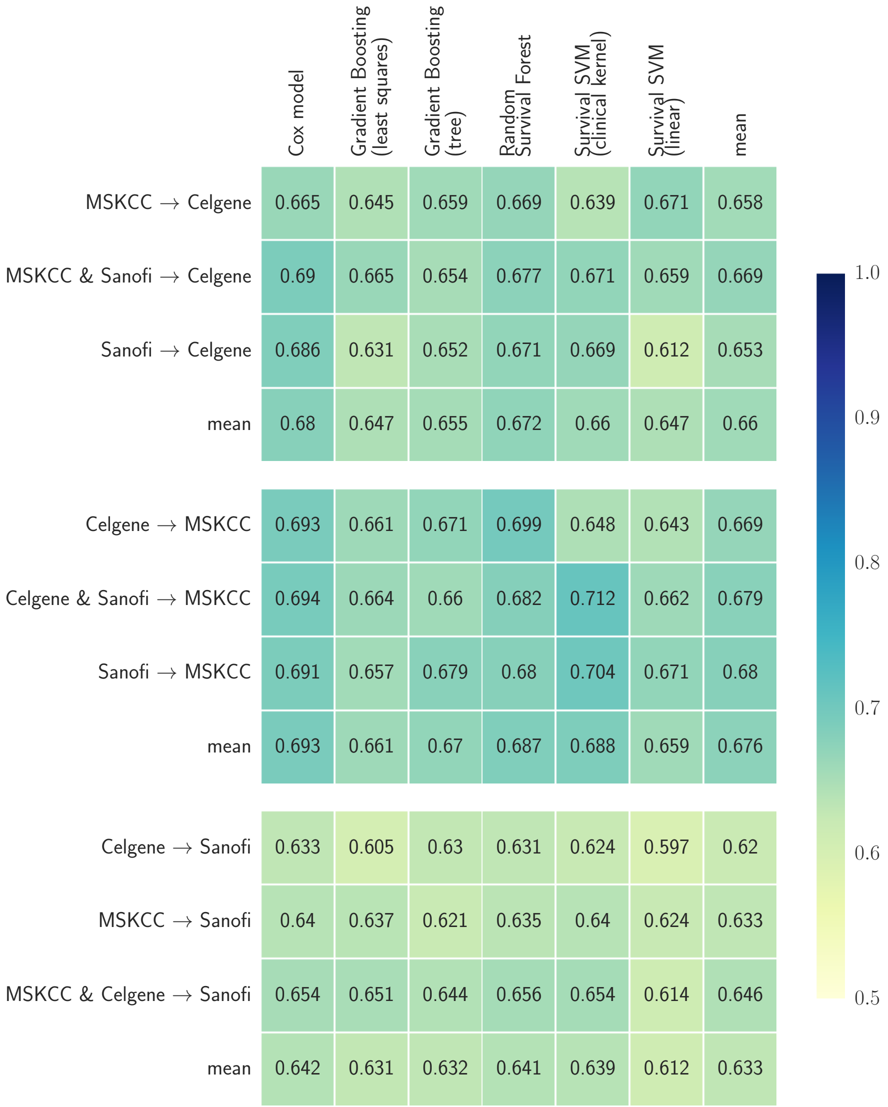
Finally, I want to mention that 5 out of 6 survival models performed worst on the Sanofi data. Although it contains the largest number of patients, its median follow-up time is almost by a factor two larger than the next study and the overlap in the distribution of censoring and survival times is rather small (see fig. 7.1). Moreover, the amount of censoring in the Sanofi study is relatively low compared to the other studies. Therefore, the observed drop in performance might stem from the fact that the bias of Harrell's concordance index usually increases as the amount of censoring increases [11]. As an alternative, I considered the integrated area under the time-dependent, cumulative-dynamic ROC curve [153, 293], which was used as alternative evaluation measure in the Prostate Cancer DREAM Challenge. However, comparing the estimated integrated area under the ROC curve across multiple datasets is not straightforward when follow-up times differ largely among studies. If the integral is estimated from time points that exceed the follow-up time of almost all patients, the inverse probability of censoring weights used in the estimator (3.129) cannot be computed, because the estimated probability of censoring at that time point becomes zero. On the other hand, if time points are defined too conservatively, the follow-up period of most patients will end after the last time point and the estimator would ignore a large portion of the follow-up period.

In the training data of the Prostate Cancer DREAM challenge, the median follow-up times were 279, 357 and 642.5 days for studies from Celegne, Memorial Sloan Kettering Cancer Center and Sanofi, respectively. Hence, defining time points that lead to adequate estimates of performance in all three datasets is challenging due to large differences in the duration of follow-up periods (see fig. 7.1).

### 7.5.3 Results of Between Study Validation

In the second set of experiments, training data and testing data were from separate studies, which resembles the setup of the Prostate Cancer DREAM challenge. Figure 7.4 summarizes the results of all experiments. Note that the number of features considered in these experiments corresponded to the intersection between features of training and test data.

Overall, the performance of models was in a similar range as in the previous set of experiments, except if Sanofi data was used for testing. If performance was estimated on the Sanofi data, models performed considerably worse compared to the remaining datasets. I believe the reason for these results are similar to the cross-validation results on the Sanofi data described in the previous section. The bias of Harrell's concordance index likely is one factor, the other that the follow-up times differed drastically between training and testing in this setting. If the follow-up period is much shorter in the training data than in the testing data, it is likely that models generalize badly for time points that were never observed in the training data, which is only the case if the



| | Cox model | Gradient Boosting (least squares) | Gradient Boosting (tree) | Random Survival Forest | Survival SVM (clinical kernel) | Survival SVM (linear) | mean |
|---|---|---|---|---|---|---|---|
| MSKCC → Celgene | 0.665 | 0.645 | 0.659 | 0.669 | 0.639 | 0.671 | 0.658 |
| MSKCC & Sanofi → Celgene | 0.69 | 0.665 | 0.654 | 0.677 | 0.671 | 0.659 | 0.669 |
| Sanofi → Celgene | 0.686 | 0.631 | 0.652 | 0.671 | 0.669 | 0.612 | 0.653 |
| mean | 0.68 | 0.647 | 0.655 | 0.672 | 0.66 | 0.647 | 0.66 |
| Celgene → MSKCC | 0.693 | 0.661 | 0.671 | 0.699 | 0.648 | 0.643 | 0.669 |
| Celgene & Sanofi → MSKCC | 0.694 | 0.664 | 0.66 | 0.682 | 0.712 | 0.662 | 0.679 |
| Sanofi → MSKCC | 0.691 | 0.657 | 0.679 | 0.68 | 0.704 | 0.671 | 0.68 |
| mean | 0.693 | 0.661 | 0.67 | 0.687 | 0.688 | 0.659 | 0.676 |
| Celgene → Sanofi | 0.633 | 0.605 | 0.63 | 0.631 | 0.624 | 0.597 | 0.62 |
| MSKCC → Sanofi | 0.64 | 0.637 | 0.621 | 0.635 | 0.64 | 0.624 | 0.633 |
| MSKCC & Celgene → Sanofi | 0.654 | 0.651 | 0.644 | 0.656 | 0.654 | 0.614 | 0.646 |
| mean | 0.642 | 0.631 | 0.632 | 0.641 | 0.639 | 0.612 | 0.633 |

**Figure 7.4**: Performance results using hold-out data from Memorial Sloan Kettering Cancer Center (MSKCC), Celgene and Sanofi. One study was used as hold-out data (indicated by the name to the right of the arrow) and one or two of the remaining studies as training data. Numbers indicate Harrell's concordance index on the hold-out data.

Sanofi data is used for testing, but not if data from Celgene or the Memorial Sloan Kettering Cancer Center is used (cf. fig. 7.1).

The experiments also confirmed observations discussed in the previous section: 1) on average, random survival forest performed better than gradient boosting and survival SVM, and 2) using survival support vector machines with clinical kernel is preferred over the linear model. Interestingly, all models, except linear survival SVM, performed best when trained on the maximum number of available patient records, which is different from results in the previous section, where models trained on data with more features performed better. Moreover, an unexpected result is that Cox's proportional hazards model was able to outperform many of the machine learning methods, including random survival forest, which is able to implicitly model non-linear relationships that are not considered by Cox's proportional hazards model. The results show that models with embedded feature selection (gradient boosting and random survival forest) are not necessarily better than models that take into account all features (Cox model and survival SVM), which is orthogonal to results in the previous section. A possible explanation for this observation might be slight differences in the importance of features between training and test data and considering more features compensates for that; whereas a parsimonious model is unable to explain the test data well, as is evident from the performance of gradient boosting with componentwise least squares base learner.

#### Conclusion

From the results presented in this and the previous section, I made the following conclusions, which ultimately led to employing heterogeneous survival models:

1. survival support vector machine should be used in combination with the clinical kernel,
2. increasing the number of samples is preferred over increasing the number of features, especially if follow-up periods are large,
3. there is no single survival model that is clearly superior to all other survival models.

## 7.6 Results of Prostate Cancer DREAM Challenge

In the previous section, I described results of internal cross-validation on data from Memorial Sloan Kettering Cancer Center, Celgene and Sanofi. In this section, I will illustrate results on the independent AstraZeneca trial. As such, evaluation was performed by the Prostate Cancer DREAM challenge organizers during the final submission round to determine the winners of individual subchallenges.

### 7.6.1 Evaluation Procedure

All predictions submitted to the first subchallenge were evaluated based on the integrated area under the time-dependent, cumulative-dynamic ROC curve (see section 3.7 on page 83), integrated over time points every 6 months up to 30 months after the first day of treatment. In addition, models that predict the exact time of death were evaluated based on the root mean squared error (RMSE) with respect to deceased patients in the test data.

Instead of relying on a single performance estimate, the challenge's final evaluation was based on one thousand bootstrap samples of data from 313 patients from the AstraZeneca study. Based on these bootstrap samples, several additional quantities were computed, a $p$-value that tells whether a prediction is significantly different from random, a Bayes factor that indicates the strength of evidence that a submission is better than the model by Halabi *et al.* [121], and a Bayes factor that compares a model to the overall best performing model.

**Hypothesis Test.** The $p$-value was obtained via permutation testing (see e.g. [115]). The distribution of a performance score under the null hypothesis that a model's predictions are no different from a random model is constructed by randomly permuting the labels in the test data and re-computing the performance of a model's predictions. By repeating this process several times, a one-sided $p$-value can be calculated as the proportion of permutations where the estimated performance was higher than on the original test data. Let $f$ denote a model, $\mathcal{D}_0$ the original test data and $\tilde{\mathcal{D}}_i$ the $i$-th permuted test data. A $p$-value can be determined empirically by

$$p = \frac{1}{k} \sum_{i=1}^{k} I(\operatorname{perf}(f, \mathcal{D}_0) \leq \operatorname{perf}(f, \tilde{\mathcal{D}}_i)), \tag{7.1}$$

where $\operatorname{perf}(f, \mathcal{D})$ is a function that computes a performance measure of a model $f$ on data $\mathcal{D}$ [115]. In the challenge's context, $k = 1000$ and perf corresponded either to the integrated area under the time-dependent ROC curve or the negative root mean squared error.

**Bayes Factor.** The Bayes factor provides an alternative to traditional hypothesis testing, which relies on $p$-values (see e.g. [311]). Given two models $f_1$ and $f_2$ and an observed performance measure $\rho$ of model $f_1$, the Bayes factor $B_{12}$ is defined as

$$B_{12} = \frac{P(\rho|f_1)}{P(\rho|f_2)} = \frac{\int P(\rho|\theta)P(\theta|f_1)d\theta}{\int P(\rho|\theta)P(\theta|f_2)d\theta}, \tag{7.2}$$

where $P(\rho|f_1)$ is the likelihood under model $f_1$ and $P(\theta|f_1)$ the prior distribution of $\theta$, with $\theta$ denoting the true performance of model $f_1$ (analogous for $P(\rho|f_2)$ and $P(\theta|f_2)$). Equation (7.2) shows that the Bayes factor is the ratio of the probability of observing $\rho$ under each of the models. In Bayesian terms, posterior beliefs (posterior odds) are formed by multiplying prior beliefs (prior odds) by the Bayes factor [311]:

$$\frac{P(f_1|\rho)}{P(f_2|\rho)} = B_{12} \frac{P(f_1)}{P(f_2)}.$$

Thus, the Bayes factor is a measure of evidence: the larger the deviation from 1, the stronger the evidence that prior beliefs must be updated. A common scale to interpret the evidence is due to Jeffreys [160] and summarized in table 7.3.

**Table 7.3**: Scale of evidence for Bayes factors as suggested in [160].

| $B_{12}$ | Interpretation |
|---|---|
| $< 10^{-1}$ | Strong evidence for $f_2$. |
| $[10^{-1}; 3^{-1}]$ | Moderate evidence for $f_2$. |
| $[3^{-1}; 1]$ | Weak evidence for $f_2$. |
| $[1; 3]$ | Weak evidence for $f_1$. |
| $[3; 10]$ | Moderate evidence for $f_1$. |
| $> 10$ | Strong evidence for $f_1$. |

In the Prostate Cancer DREAM challenge, the Bayes factor was used to determine if there is sufficient evidence to argue that one model is superior or inferior to another model. Models were considered as tied if the associated Bayes factor only indicated weak evidence according to table 7.3. The Bayes factor was estimated from $k = 1000$ bootstrap samples $\mathcal{D}_i$ of the test data $\mathcal{D}_0$ by

$$\hat{B}_{12} = \frac{\sum_{i=0}^{k} I(\mathrm{perf}(f_1, \mathcal{D}_i) > \mathrm{perf}(f_2, \mathcal{D}_i))}{\sum_{i=0}^{k} I(\mathrm{perf}(f_1, \mathcal{D}_i) \leq \mathrm{perf}(f_2, \mathcal{D}_i))}. \tag{7.3}$$

### 7.6.2 Results of Subchallenge 1a: Ranking Patients According to Survival Time

Based on conclusions drawn from experiments on the challenge's training data in section 7.5, the final model was chosen to be a heterogeneous ensemble of different survival models, trained on the combined data from all three studies to maximize the amount of patients in the training data. Four of the models listed in section 7.5.1 formed the basis of the ensemble; linear survival support vector machines were excluded, because they performed poorly when not combined with the clinical kernel. Cox's proportional hazards model had to be excluded, because I encountered numerical problems in the optimization that could not be resolved before the conclusion of the challenge. Due to all survival models having one or more hyper-parameters, I added the same model multiple times, but each with a different hyper-parameter configuration, as summarized in table 7.4 and described in more detail in appendix B.2. The vast majority (1,728) of models in the library corresponded to gradient boosting with regression trees as base learners, because regression trees had the most hyper-parameters. In total, the library comprised 1,801 models. Performance was estimated based on cross-validated models with five folds (see algorithm 7.1) and Harrell's concordance index [127, 128].

After estimating the performance of all 1,801 models on the combined data from all three studies, models with a $c$ index below 0.66 were discarded, which left 999 candidate models to be evaluated during prediction. During prediction, the diversity of

**Table 7.4**: Heterogeneous ensemble of survival models used in subchallenge 1a of the Prostate Cancer DREAM challenge. *All* denotes the initial size of the ensemble, *Pruned* the size after pruning models with Harrell's concordance index below 0.66, and *Top 5%* to the final size of the ensemble corresponding to the top 5% according the combined accuracy and diversity score in algorithm 7.2.

| | Configurations | | |
|---|---|---|---|
| Survival Model | All | Pruned | Top 5% |
| Gradient Boosted Cox Model (tree) | 1,728 | 936 | 56 |
| Gradient Boosted Cox Model (least squares) | 36 | 36 | 7 |
| Random Survival Forest | 24 | 24 | 24 |
| Ranking-based Survival SVM (clinical kernel) | 13 | 3 | 3 |
| $\sum$ | 1,801 | 999 | 90 |

all remaining models was evaluated following algorithm 7.2, where Pearson's correlation coefficient and $\tau_{\text{corr}} = 0.6$ was used. The final ensemble was limited to the top 5% (90) models based on the combined score of concordance index and diversity as summarized in table 7.4.

Figure 7.5 depicts scatter plots comparing models' performance and diversity. Most of the gradient boosting models with regression trees as base learners were pruned because their predictions were redundant to other models in the ensemble (upper left). In contrast, all random survival models remained in the ensemble throughout (lower left). The highest diversity was observed for gradient boosting models (mean = 0.279) and the highest accuracy for random survival forests (mean = 0.679). The final ensemble comprised all types of survival models in the library, strengthening my conclusion from experiments in section 7.5.1 that there is no universally best survival model.

Figure 7.6 summarizes the performance of submitted models by all teams, evaluated on the challenge's test data consisting of 313 patients of the AstraZeneca study. As expected, all models performed significantly better than random and 30 out of 51 models outperformed the baseline model by Halabi *et al.* [121] by achieving a Bayes factor greater than 3. Results show that there was a clear winner in team FIMM-UTU and that the performance of the remaining models were very close to each other; there was merely a difference of 0.0171 points in integrated area under the ROC curve (iAUC) between ranks 2 and 25.

The proposed heterogeneous ensemble of survival models by team CAMP achieved an iAUC score of 0.7646 on the test data and was ranked 23rd according to iAUC and 20th according to Bayes factor with respect to the best model (FIMM-UTU). When considering the Bayes factor of the proposed ensemble method to all other models, there is only sufficient evidence (Bayes factor greater 3) that five models performed better (marked italic bold in fig. 7.6). The Bayes factor to the top two models was

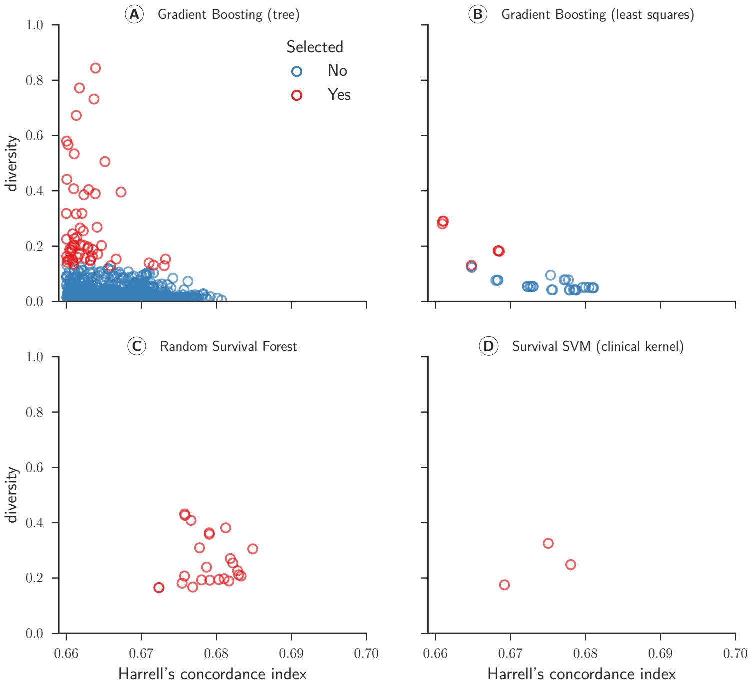


**Figure 7.5**: Concordance index and diversity score of 999 survival models for subchallenge 1a. The concordance index was evaluated by cross-validated models on the training data from the Memorial Sloan Kettering Cancer Center, Celgene and Sanofi. Diversity was computed based on Pearson's correlation coefficient between predicted risk scores for 313 patients of the AstraZeneca trial (final scoring set).

20.3 and 6.6 and ranged between 3 and 4 for the remaining three models. With respect to the model by Halabi *et al.* [121], there was strong evidence (Bayes factor 12.2; iAUC 0.7432) that heterogeneous ensembles of survival models could predict survival of mCRPC patients more accurately.

## 7.6.3 Results of Subchallenge 1b: Predicting Exact Time of Death

In the second part of subchallenge one, participants were tasked with predicting the exact time of death rather than ranking patients according to their survival time. As for the first part of this subchallenge, the final model was a heterogeneous ensemble, but based on a different library of models. Three models formed the basis of the ensemble: gradient boosted accelerated failure time model [148] with randomized regression trees [37, 39] or componentwise least squares as base learner [45], and hybrid survival support vector machine (5.37) with clinical kernel [70, 236]. The library contained several of these models with different hyper-parameter configurations as summarized in table 7.5 (see appendix B.2 for a complete list). The ensemble was constructed in a similar manner as the ensemble used for subchallenge 1a. Instead

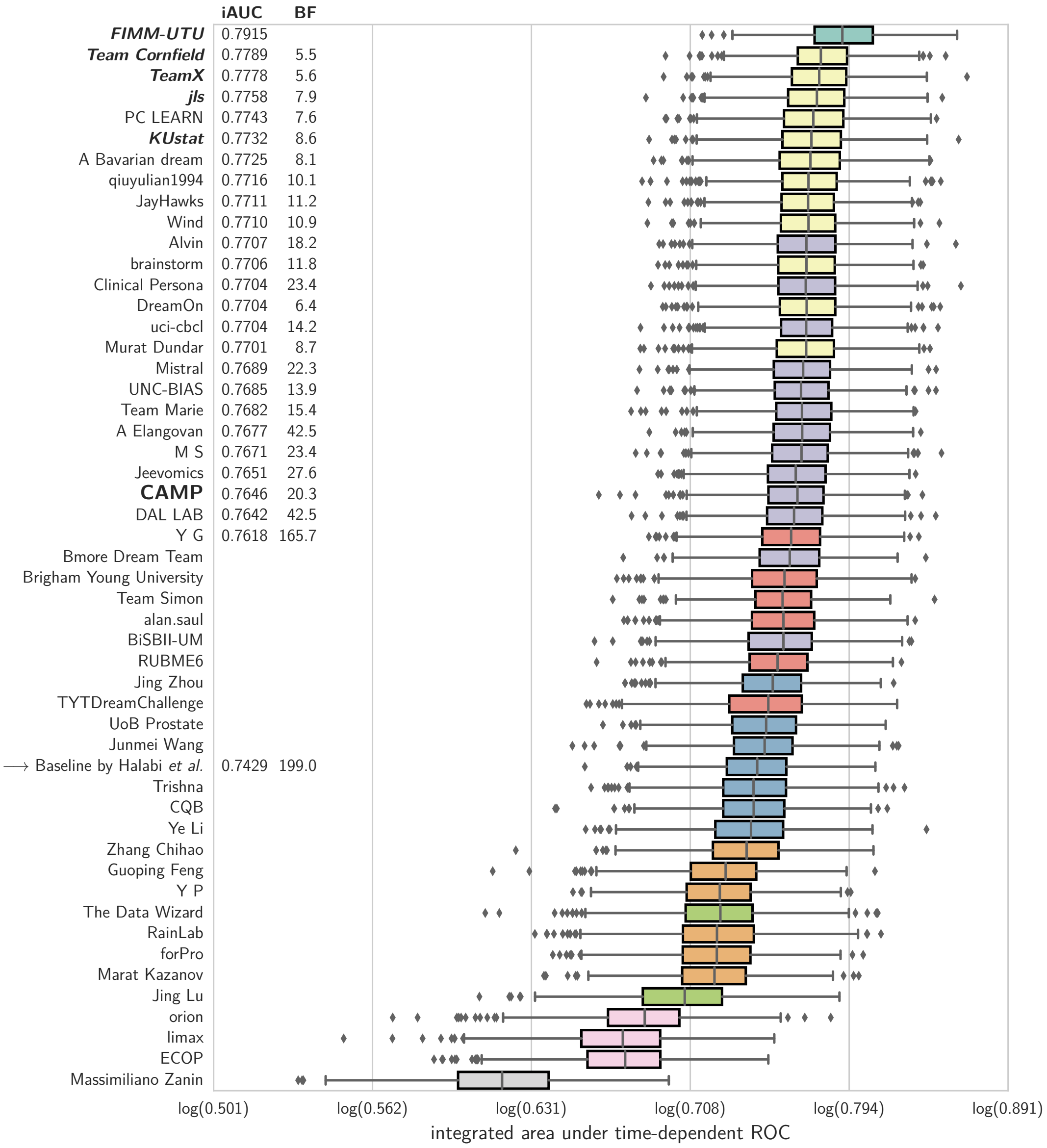


**Figure 7.6**: Final results of all 51 teams for subchallenge 1a. Models are sorted according to the integrated area under the time-dependent, cumulative-dynamic ROC curve (iAUC) on 313 patients of the AstraZeneca trial. Colors indicate groups of models that performed similar to the top-performing model in that group according to Bayes factor (weak evidence according to table 7.3). Submissions in bold italic performed better than the proposed heterogeneous ensemble of survival models (moderate or strong evidence). The baseline model by Halabi *et al.* [121] is marked with an arrow. CAMP: Heterogeneous ensemble of survival models. BF: Bayes factor of top-performing model compared to other models. Figure is based on data courtesy of Tao Wang and the Prostate Cancer DREAM challenge organizers.

**Table 7.5**: Ensemble used in subchallenge 1b. *All* denotes the initial size of the ensemble, *Pruned* the size after pruning models with a root mean squared error more than 15% above the error of the best performing model, and *Top 5%* to the final size of the ensemble corresponding to the top 5% according the combined accuracy and diversity score in algorithm 7.2. AFT: Accelerated Failure Time.

| | Configurations | | |
|---|---|---|---|
| Regression Model | All | Pruned | Top 5% |
| Gradient Boosted AFT model (tree) | 1,728 | 1,236 | 90 |
| Gradient Boosted AFT model (least squares) | 36 | 36 | 0 |
| Hybrid Survival SVM (clinical kernel) | 78 | 9 | 2 |
| $\sum$ | 1,842 | 1,281 | 92 |

of concordance index, I used the root mean squared error (RMSE) with respect to all uncensored samples in the test data and converted it to an accuracy measure by multiplying its reciprocal by the RMSE of the best performing model in the library, i.e., $\text{accuracy}(i) = (\min_{j=1,\ldots,S} \text{RMSE}(j))/\text{RMSE}(i)$, where $S$ is size of the library. In addition, pruning by accuracy and diversity was performed during training following algorithms 7.1 and 7.2 using Pearson's correlation coefficient between models' residuals and setting $c_{\text{min}} = 0.85$ as well as $\tau_{\text{corr}} = 0.6$.

After evaluating the RMSE of all 1,842 models in the ensemble, models with an RMSE more than 15% above the RMSE of the best performing model in the library were discarded. The diversity score from the remaining 1,281 models was computed and added to the accuracy score according to algorithm 7.2. The final ensemble comprised only regression models that were among the top 5% (92) with respect to the combined accuracy-diversity score. Individual pruning steps are summarized in table 7.5.

Figure 7.7 illustrates the RMSE and diversity of all models after the first pruning step. In contrast to the ensemble of survival models used in the first part of subchallenge one, the ensemble in this subchallenge was characterized by very little diversity: the highest diversity was 0.064. In fact, all 92 models included in the final ensemble had a diversity score below 0.001, which means that pruning was almost exclusively based on the RMSE. Gradient boosting models with componentwise least squares base learners were completely absent from the final ensemble and only two hybrid survival support vector machine models had a sufficiently low RMSE to be among the top 5%.

The evaluation of all submitted models on the challenge's final test data from the AstraZeneca trial is summarized in fig. 7.8. In this subchallenge, the proposed heterogeneous ensemble of regression models achieved the lowest root mean squared error (194.4) among all submissions. Similar to the results in the first part, the difference in RMSE between the 1$^{\text{st}}$ placed model and the 25$^{\text{th}}$ placed model was less than 25. With respect to my proposed winning model, there was insufficient evidence to state it

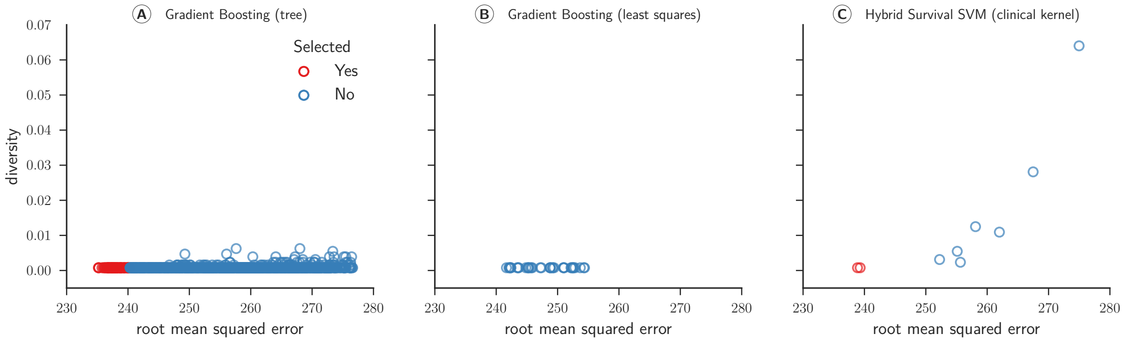


**Figure 7.7**: Root mean squared error (RMSE) and diversity score of 1,281 regression models for subchallenge 1b. The RMSE was evaluated by cross-validated models on the training data from the Memorial Sloan Kettering Cancer Center, Celgene and Sanofi. Diversity was computed based on Pearson's correlation coefficient between residuals on the training data.

outperformed all other models, because the comparison to five other models yielded a Bayes factor less than three.

## 7.6.4 Discussion and Conclusion

In the Prostate Cancer DREAM challenge, participants were asked to develop novel approaches to predict survival of metastatic, castrate-resistant prostate cancer patients based on health records of 1,600 patients, combined from three phase III clinical trials. To successfully complete this task, teams had to develop an end-to-end solution that addresses several intermediate steps, from extracting patient-level features from raw data, over imputing missing values, transforming and cleaning data, to eventually training a predictive survival model. Teams' final solutions were characterized by a large diversity due to countless alternatives for each of these steps. Nevertheless, results indicate that the predictive performance of most proposed solutions were statistically indistinguishable from each other, despite a large variety in the choice of features and algorithms that led to the final prediction.

My proposed solution was mostly based on data-driven techniques without including much prior knowledge about prostate cancer in general. Of course, this is partly attributed to the lack of detailed medical background in the field of prostate cancer, which I had none before working on this challenge. In the first step, features were derived from raw medical records by extracting as much information as possible such that a patient's state can be described accurately. Due the data being a combination of four clinical trials, several issues were identified early on: 1) features contained structured missingness, because some information was not recorded for a subset of trials, 2) the duration of follow-up periods differed considerably among trials, and 3)

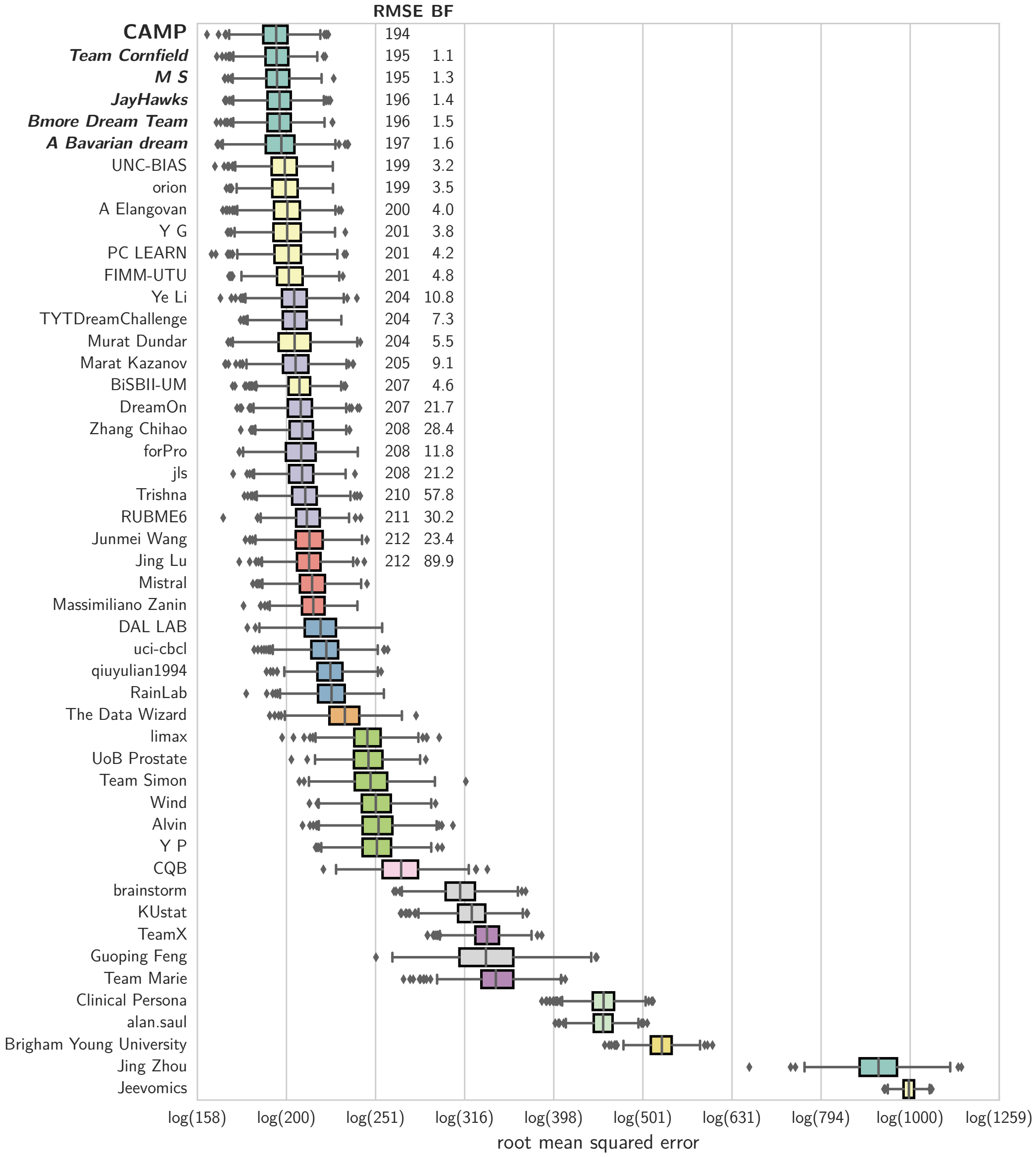


**Figure 7.8**: Final results of all 51 teams for subchallenge 1b. Models are sorted according to the root mean squared error (RMSE) on 313 patients of the AstraZeneca trial. Colors indicate groups of models that performed similar to the top-performing model in that group according to Bayes factor (weak evidence according to table 7.3). Submissions in bold italic performed similar to the proposed heterogeneous ensemble of regression models (weak evidence). CAMP: Heterogeneous ensemble of regression models. BF: Bayes factor of top-performing model compared to other models. Figure is based on data courtesy of Tao Wang and the Prostate Cancer DREAM challenge organizers.

studies used a slightly different nomenclature in describing medications, therapies and comorbitities.

I addressed the first issue by partitioning the data into seven subsets based on the extent of available features for a particular study, or the combination of data from multiple studies (see section 7.2). Regarding the second issue, an extensive set of experiments summarized in section 7.5 demonstrated that if the follow-up period of patients used during training is much shorter than for patients in the test data, all survival models generalize modestly, disregarding the amount of features available during training. From these results, I concluded that it would be best to combine data from all three clinical trials for the final prediction in order to maximize the number of distinct time points a model has seen during training. Interestingly, the winning team of subchallenge 1a completely excluded data from the Memorial Sloan Kettering Cancer Center in their solution. They argued that it was too dissimilar to data of the remaining three studies, including the test data [181]. Therefore, it would be interesting to investigate unsupervised approaches proposed in chapter 6 that could deduce a similarity or distance measure between patients. The resulting similarities could be used to decrease the influence of outlying patients during training.

The second important conclusion from experiments in section 7.5 is that no survival model clearly outperformed all other models in all the evaluated scenarios. To some extent, this not surprising since the “no free lunch” theorem already states that the average performance of any two algorithms averaged over all problems is equal [319, 320]. The theorem implies that if one algorithm outperforms another algorithm on one problem, there must be a different problem were the relationship is reversed. Therefore, instead of relying on a single survival model with a single hyper-parameter configuration, I constructed a heterogeneous ensemble of several survival models with different hyper-parameter configurations. In total, I considered a library consisting of over 1,800 different models, which was pruned to ensure accuracy and diversity of models as described in section 7.4.

The proposed ensemble approach was able to secure the win in subchallenge 1b, where the task was to predict the exact time of death rather than providing a relative risk score, which was the objective of subchallenge 1a. In subchallenge 1a, the ensemble approach was significantly outperformed by models of five competing teams (see fig. 7.6). Due to large differences in teams’ overall solutions it is difficult to pinpoint the reason for the observed performance difference: it could be attributed to the choice of predictive model, but also to choices made during pre-processing or filtering the data. From my experience of the three intermediate scoring rounds before the final submission, I would argue that identifying the correct subset of patients in the training data that is most similar to the test data is more important than choosing a predictive model. By training a survival model on data combined from three trials and applying it to patients from a fourth trial, inconsistencies between studies inevitably lead to outliers with respect to the test data, which in turn diminishes the performance of a model – if not addressed explicitly during training.

A possible explanation why the heterogeneous ensemble worked better for regression than for survival analysis might be how predictions from members of the ensemble were aggregated. In regression, the prediction is a continuous value that directly corresponds to the time of death, which allows simple averaging of individual predictions. I used the same approach to average predictions from survival models, despite slightly different semantics between predictions of regression and survival models. Although both predictions are real-valued, the prediction of a survival model does generally not correspond to the time of death, but is a risk score on an arbitrary scale. A homogeneous ensemble only consists of models of the same type, therefore predictions can be aggregated by simply computing the average. A problem arises for heterogeneous ensembles if the scale of predicted risk scores differs among models.

To illustrate the problem, consider an ensemble consisting of survival trees as used in a random survival forest (see section 3.6 on page 79) and ranking-based linear survival support vector machines (see chapter 5 on page 99). The prediction of the former is based on the cumulative hazard function estimated from samples residing in leaf nodes a new sample was assigned to. Thus, predictions are always positive due to the definition of the cumulative hazard function in eq. (2.8). In contrast, the prediction of a linear survival SVM is the inner product between a model's vector of coefficients and a sample's feature vector, which can take on negative as well as positive values. It is easy to see that, depending on the scale difference, simply averaging predicted risk scores favors models with generally larger risk scores (in terms of absolute value) or positive and negative predicted risk scores cancel each other out. Instead of simply averaging risk scores, the problem could be alleviated if models' risk scores were first transformed into ranks, thereby putting them on a common scale, before averaging the resulting ranks.

## Conclusion

I proposed an end-to-end solution to predict survival of metastatic, castrate-resistant prostate cancer patients based on features derived from medical records. In my opinion, the most challenging part in the Prostate Cancer DREAM Challenge was to find a suitable way to combine training data from three separate clinical trials such that models still generalized well when applied to patients from an independent fourth study. The main focus of the challenge was to develop new models that predict a patient's relative risk of death. I addressed this problem by proposing heterogeneous survival ensembles, which are able to aggregate predictions from a wide variety of survival models. Although the model was significantly outperformed by 5 out of 50 competing solutions in subchallenge 1a, the proposed ensemble approach for subchallenge 1b could predict the exact time of death more accurately than any other submitted model. I believe this result is encouraging and warrants further research in using heterogeneous ensembles for survival analysis.

The code and documentation underlying the methods presented in this chapter can be found at `http://dx.doi.org/10.7303/syn3647478`.

# 8 Conclusion

I have addressed several important questions related to survival analysis using machine learning techniques. First, I have introduced a much improved optimization algorithm for linear ranking-based survival support vector machines. I have shown that training is characterized by a lower time and space complexity than previous training algorithms, without drawing upon an approximation of the objective function. Second, I demonstrated that the same ideas can be used to obtain a non-linear decision boundary by applying the representer theorem and performing optimization in the primal rather than the dual. Moreover, it is straightforward to obtain a hybrid model that optimizes a ranking loss and regression loss concurrently.

I studied feature extraction algorithms in the context of survival analysis with heterogeneous, high-dimensional feature vectors. I proposed utilizing random survival forests to address two of the main problems encountered with feature extraction methods based on spectral embedding: 1) neighborhood graph construction and 2) out-of-sample extension. In addition, using a random survival forest to construct a neighborhood graph offers the advantage that right censored survival times are taken into account. I empirically evaluated 10 combinations of feature extraction methods and 8 survival models, which led me to conclude that a survival model trained on a low-dimensional embedding of the training data is a valuable alternative to a survival model with embedded feature selection if the number of training samples is sufficient to describe the underlying manifold. For small sample sizes ($< 500$), embedded feature selection methods are preferred.

In my contributions to the Prostate Cancer DREAM Challenge, I introduced heterogeneous survival ensembles that build upon the diversity in available survival models. When predicting the exact time of death of patients with metastatic, castrate-resistant prostate cancer, my heterogeneous ensemble of gradient boosted accelerated failure time models and hybrid survival support vector machines achieved the lowest prediction error among all 51 submissions.

There are other research questions I have not presented in this dissertation. One advantage of kernelized survival support vector machines is that it could be applied to any kind of data as long as similarities can be captured by a suitable positive definite kernel matrix. This offers interesting applications when training samples are represented as structured objects rather than feature vectors. For instance, the gene ontology [14] represents genes in an acylic graph according to the function of associated proteins. Thus, instead of merely using gene expression values as a vector, prior information could be incorporated into the learning process by encoding expression values in a graph representing genes' biological functions. A similar concept was used in [330], where prior knowledge about co-expression of genes was encoded in a graph and used in a regularizer to Cox's proportional hazards model. However, samples are still presented as feature vectors and only the corresponding coefficients are constrained by the graph. Graph-based representations are also common outside of omics data: medical conditions are organized in the International Classification of Diseases (ICD), treatments are organized in diagnosis-related groups, and RadLex [187] provides a hierarchical nomenclature of radiology terms.

If a patient's state is described by multiple structured objects, such as the ones described above, another interesting problem arises. In the literature, this problem is often referred to as multi-modality learning, which is often studied when combining imaging data (e.g. ultrasound and computer tomography) from multiple imaging modalities (e.g. [143, 289, 324]). If similarities between subjects are described by modality-specific kernel matrices, multiple kernel learning can be used to obtain a kernel matrix that fuses information from all modalities (e.g. [185, 274]). The resulting kernel matrix is a weighted sum of modality-specific kernels and weights are learned during training. The motivation of multiple kernel learning is similar to my motivation of using multiview spectral embedding to combine information from multiple sources in chapter 6. To my surprise, experiments did not show any apparent advantage of multiview spectral embedding over singleview spectral embedding. It would be interesting to investigate whether this observation holds for multiple kernel learning as well.

Finally, I investigated multi-task learning for survival analysis. Multi-task learning algorithms explicitly model the scenario when observations in a dataset are clustered into tasks and the relationship between features and survival time slightly differ from task to task. For instance, consider a dataset collected from multiple centers. Usually, such data will feature center-specific effects due to different protocols, treatments, patient population, and so forth [8] – something I observed in the context of the Prostate Cancer DREAM Challenge in chapter 7 too. A single model that does not distinguish between centers (the tasks) would likely generalize badly, because estimated coefficients are biased. In contrast, if a model is fit to data of each individual center, the sample size would be rather small and lead to overfitting in the sense that the model is ineffective when applied to data from a different center.

Multi-task learning tackles this problem by finding latent commonalities among centers and modeling the center-specific effects that cannot be explained by a common model. Formally, for an arbitrary loss function $L$ and $T$ tasks, the multi-task learning objective is

$$\underset{\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_T}{\operatorname{argmin}} \quad \sum_{i=1}^{\top} L(\boldsymbol{w}_0 + \boldsymbol{w}_t, \mathcal{D}_t) + \sum_{t=0}^{\top} \lambda_t \|\boldsymbol{w}_t\|_2^2, \tag{8.1}$$

where $\lambda_t > 0$ is a task-specific regularization term, $\boldsymbol{w}_0$ are the coefficients representing the commonality, and $\boldsymbol{w}_t$ and $\mathcal{D}_t$ the coefficients and training data of the $t$-th task, respectively. I experimented with this model by following techniques proposed by Argyriou *et al.* [12] and Chapelle *et al.* [55]. Preliminary results using synthetic as well as data from the Prostate Cancer DREAM Challenge showed promising results. Multi-task survival models outperformed aggregate models (a single model for all tasks) and independent models (one model for each task). However, a more elaborate evaluation concerning the interplay between the number of tasks and number of samples and its effect on the predictive performance is necessary. In addition, similar ideas have been explored in statistics, where the commonality is modeled as a fixed effect and the impact of individual tasks as random effect. With respect to survival analysis, this is known as a frailty model (e.g. [80]).

I believe that the ongoing surge of medical data requires the development of novel machine learning techniques to maintain the advancement of medical science, which is expected to draw conclusions from massive amounts of clinical data. In this dissertation, I contributed to the solution of this challenge by proposing new ideas to improve learning from large, heterogeneous survival data. In particular, I demonstrated that state-of-the-art methods in convex optimization significantly reduce training time and space requirements of existing survival models, and that it is necessary to explicitly consider survival time, censoring and that feature vectors are a mix of continuous and categorical variables when applying feature extraction methods to overcome the curse of dimensionality.

# Appendix A

# Experiments Regarding Dimensionality Reduction Methods

## A.1 Hyper-Parameter Configurations

The list below provides a detailed list of hyper-parameter configurations used in the comparison of feature extraction and feature selection methods in section 6.3.

$*$: Only relevant if neighborhood graph was constructed by constraining edges between patients of similar survival time. †: Only relevant if stochastic gradient boosting was performed. ‡: Only relevant if gradient boosting with dropout was performed.

- Multview Spectral Embedding (40 configurations or 160 configurations$^*$):
  - Complementary factor $r$: 1.1, 1.3, 1.5, 2, 6
  - Dimensionality $d$ of low-dimensional representation (relative to full data): 1%, 5%, 10%, 15%, 20%, 25%, 50%, 75%
  - Number of nearest neighbors: log(# samples)
  - Percentiles$^*$: $[50]$; $[33, 66]$; $[25, 50, 75]$; $[20, 40, 60, 80]$
- Laplacian Eigenmaps and Locality Preserving Projections (8 configurations):
  - Dimensionality $d$ of low-dimensional representation (relative to full data): 1%, 5%, 10%, 15%, 20%, 25%, 50%, 75%
  - Number of nearest neighbors: log(# samples)
- Principal Component Analysis (8 configurations):
  - Dimensionality $d$ of low-dimensional representation (relative to full data): 1%, 5%, 10%, 15%, 20%, 25%, 50%, 75%
- Kernel PCA (8 configurations):
  - Dimensionality $d$ of low-dimensional representation (relative to full data): 1%, 5%, 10%, 15%, 20%, 25%, 50%, 75%
  - Kernel function: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\mathrm{d}_{\mathrm{RSF}}(\boldsymbol{x}_i, \boldsymbol{x}_j))$
- Cox's Proportional Hazards Model with $\ell_1$ (LASSO) or $\ell_2$ (ridge) penalty (13 configurations):

  - Regularization weight $\lambda$: $2^{-12}, 2^{-10}, \ldots, 2^{12}$

- Survival Support Vector Machine (13 configurations):
  - Weight $\gamma$ of loss function: $2^{-12}, 2^{-10}, \ldots, 2^{12}$
- Random Survival Forest (24 configurations):
  - Number of trees: 1,000
  - Number of features to evaluate per split: $\sqrt{\#\text{ features}}$
  - Number of candidate splits to evaluate per feature: 2, 5, 10, $\infty$
  - Minimum number of samples in a terminal node: 3, 5, 10, 25, 50, 100
- Gradient boosting with regression tree as base learner (1,728 configurations$^\dagger$ or 2,304 configurations$^\ddagger$):
  - Number of iterations: 100, 500, 1000, 1500
  - Subsampling percentage: 100%, 75%, 50%
  - Learning rate$^\dagger$: 0.06, 0.125, 0.25
  - Dropout rate$^\ddagger$: $10^{-4}$, 0.015, 0.03, 0.045
  - Maximum number of leaf nodes: 5, 10, 20
  - Minimum number of samples per split: 2, 5, 10, 20
  - Maximum number of features to evaluate per split: all, $\sqrt{\#\text{ features}}$, 50%, 75%
- Gradient boosting with componentwise least squares as base learner (36 configurations$^\dagger$ or 48 configurations$^\ddagger$):
  - Number of iterations: 100, 500, 1000, 1500
  - Subsampling percentage: 100%, 75%, 50%
  - Learning rate$^\dagger$: 0.06, 0.125, 0.25
  - Dropout rate$^\ddagger$: $10^{-4}$, 0.015, 0.03, 0.045

## A.2 Description of Datasets

Below is a list of features and their corresponding view from three clinical datasets used in the experiments in section 6.3.

**Table A.1**: Views and their features of the breast cancer dataset [77].

| View | Feature |
|---|---|
| Demographics | Age<br>Diameter of tumor (in mm)<br>Histopathological grading<br>Estrogen-receptor-positive tumor |
| Gene expression | 76-gene signature according to Desmedt *et al.* [77] |

**Table A.2**: Views and their features of the coronary artery disease dataset [217].

| View | Feature |
|---|---|
| Angiographic measurements<br>(31 features) | # of Lesions<br>AHA/ACC class (max)<br>—— (min)<br>—— (mode)<br>Angulation (min)<br>—— (mode)<br>Bifurcation (max)<br>—— (min)<br>—— (mode)<br>Calcification (min)<br>—— (mode)<br>Chronic occlusion (min)<br>—— (mode)<br>Diameter stenosis (%; mean)<br>Eccentricity (max)<br>—— (min)<br>—— (mode)<br>Lesion in acute coronary syndrome (max)<br>—— (mode)<br>Lesion length (mm; mean)<br>Minimal luminal diameter (mm; mean)<br>Multivessel disease<br>Ostial location (max)<br>—— (mode)<br>Reference vessel diameter (mm; mean)<br>Restenotic lesion (min)<br>—— (mode)<br>TIMI flow grade before PCI (min)<br>—— (mode)<br>Tortuous vessel (min)<br>—— (mode)<br>Vessel treated (mode) |
| Laboratory biomarkers<br>(4 features) | Creatinine (mg/dl)<br>C-reactive protein (mg/l)<br>High-sensitivity troponin T ($\mu$g/l)<br>N-terminal pro–brain natriuretic peptide (ng/l) |
| Extent of disease<br>(7 features) | Angina class<br>Extent of coronary artery disease<br>Heart rate<br>Left ventricular ejection fraction (%)<br>New York Heart Association class<br>Reduced left ventricular function<br>ST-elevation myocardial infarction |
| Medications<br>(6 features) | ACE inhibitor<br>Acetylsalicylic acid<br>$\beta$-Blocker<br>Calcium antagonist |

Continued on next page

| View | Feature |
|---|---|
| | Diuretics<br>Nitrate<br>Statin |
| Demographics/Disease history<br>(12 features) | Age (years)<br>Arterial hypertension<br>Body-mass index (kg/m$^2$)<br>Diabetes<br>Family history of CAD<br>Hypercholesterolemia<br>Male<br>Previous coronary artery bypass grafting<br>Previous percutaneous coronary intervention<br>Previous myocardial infarction<br>Repeated revascularization<br>Smoking |

**Table A.3**: Views and their features of the Framingham Offspring dataset [166]. All features were collected from two exams, except those marked by *, which were only available for one exam.

| View | Feature |
|---|---|
| Demographics/Disease history<br>(24 features) | Age<br>Ankle Edema<br>Chest Discomfort*<br>Diastolic Blood Pressure (by physician)<br>Diastolic Blood Pressure (by nurse)<br>Dyspnea Increase<br>Dyspnea on Exertion<br>Orthopnea*<br>Sex*<br>Systolic Blood Pressure (by 1st physician)<br>Systolic Blood Pressure (by 2nd physician)*<br>Systolic Blood Pressure (by nurse)<br>Treatment for Hypertension<br>Weight |
| Life-style<br>(17 features) | Age Start Cigarette Smoking*<br>Beer Intake per Week<br>Cholesterol Lowering Diet<br>Cocktails per Week<br>Diabetic Diet<br>Ever Smoked Cigarettes Regularly<br>Ever Smoked Regularly<br>No. of Cigarettes per Day<br>Wine Intake per Week |
| Lipid panel<br>(25 features) | Total Plasma Cholesterol (in blood)<br>Whole Plasma Appearance (in blood)<br>Lipids Whole Plasma Pre-Beta (in blood)* |

Continued on next page

| View | Feature |
|---|---|
| | Fasting 12 Hrs<br>Fredrickson Classification<br>Pre-Beta Band*<br>Sinking Pre-Beta Band*<br>HDL Cholesterol<br>Infranate after 12 Hrs<br>LDL Cholesterol<br>Pre-Beta Bottom Fraction*<br>Pre-Beta Top Fraction*<br>Triglyceride<br>VLDL Cholesterol |
| Laboratory biomarkers<br>(32 features) | Albumin<br>Bilirubin<br>Bun<br>Calcium<br>Fasting<br>Globulin<br>HCT<br>HGB<br>LDH<br>MCH<br>MCHC<br>MCV<br>Phosphorus<br>SGOT<br>Total Protein<br>Uric Acid |
| Medication<br>(16 features) | Anti-Cholesterol Agent<br>Anti-Coagulants*<br>Bronchodilator Or Aerosol<br>Cardiac Glycosides<br>Diuretics-Hypertension<br>Hypotensive (excluding Diuretics)<br>Nitrites*<br>Thyroid<br>Tranquilizers |
| Menopause<br>(6 features) | Age Period Stopped*<br>Ever Taken Premarin*<br>Hysterectomy*<br>Oral Contraceptive*<br>Ovaries Removed*<br>Periods Stopped 1 Yr Or More* |
| Electrocardiography<br>(30 features) | Clinical Reading<br>Intraventricular Block: Bifascicular*<br>Intraventricular Block: Hemiblock<br>Intraventricular Block: Left*<br>Intraventricular Block: Right<br>Left Ventricular Hypertrophy |

Continued on next page

| View | Feature |
|---|---|
| | Myocardial Infarction* |
| | Non.specific ST-Segment Abnormality |
| | Non-specific T-Wave Abnormality |
| | Other ECG Abnormality* |
| | P-R Interval |
| | Premature Beats |
| | QRS Angle With Sign |
| | QRS Interval |
| | QT Interval |
| | Ventricular Rate |
| | Wolff-Parkinson-White Syndrome |

## A.3 View-specific Coefficients for Framingham Offspring Dataset

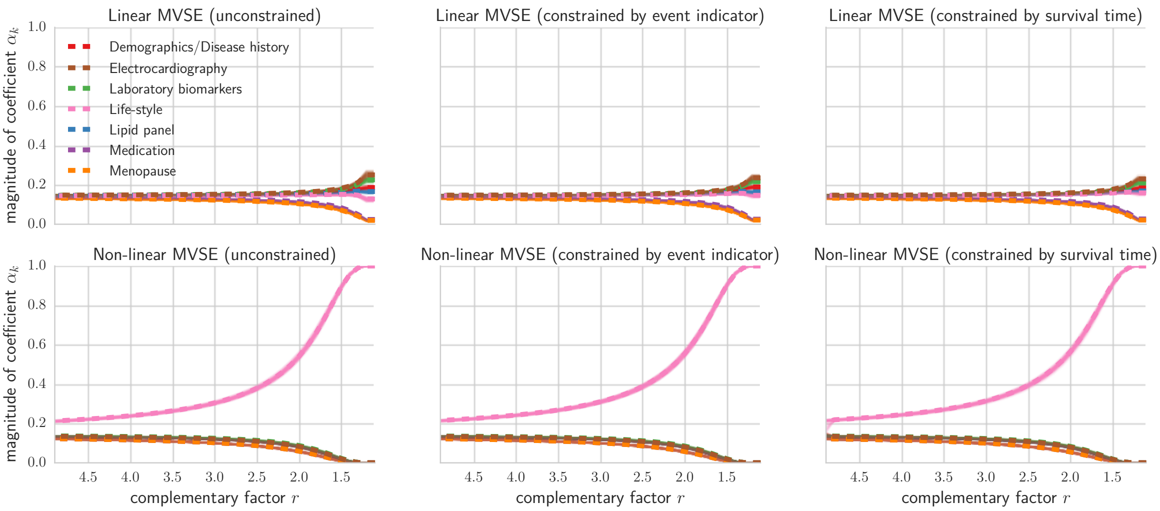


**Figure A.1**: Value of view-specific coefficients $\boldsymbol{\alpha}$ in multiview spectral embedding (MVSE) for varying complementary factor $r \in \{1.1, 1.2, \ldots, 4.9, 5\}$. Solid lines indicate the path of view-specific coefficients in 50 random subsamples of the Framingham Offspring data. The average coefficient across all subsamples is indicated by dashed lines.

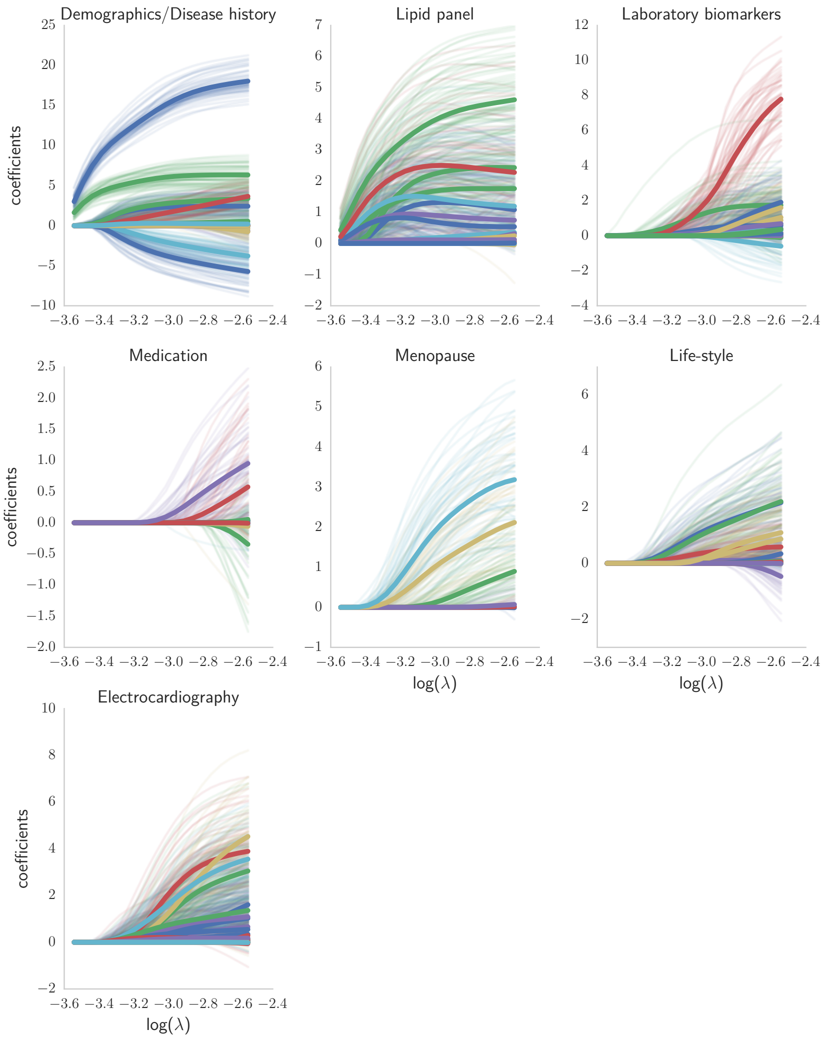


**Figure A.2**: Coefficients of Cox's proportional hazards model with group LASSO penalty. Solid lines indicate the path of coefficients across 20 fixed values of the regularization parameter $\lambda$ from 50 random subsamples of the Framingham Offspring data. The average coefficient across all subsamples is indicated by bold lines.

# Appendix B

# Prostate Cancer DREAM Challenge Data

## B.1 Features

The tables below list all features extracted from patient records from three phase III clinical trials used in the Prostate Cancer DREAM Challenge (see chapter 7): Memorial Sloan Kettering Cancer Center (MSKCC), Celgene, and Sanofi [34, 223, 283].

### B.1.1 Comorbitities

**Table B.1**: Extracted features corresponding to comorbitities. Only available for patients from Celgene or Sanofi study.

| Comorbitity |
|---|
| # of comorbidities |
| Anaemia |
| — date |
| Angina pectoris |
| Anxiety |
| — date |
| Appendicectomy |
| Arthralgia |
| — date |
| Arthritis |
| Asthenia |
| Asthma |
| Atrial fibrillation |
| Back pain |
| — date |
| Benign prostatic hyperplasia |
| Biopsy prostate |
| Bone pain |

Continued on next page

| Comorbitity |
|---|
| — date |
| Bronchitis |
| Cancer pain |
| Cataract |
| Cholecystectomy |
| Chronic obstructive pulmonary disease |
| Constipation |
| — date |
| Coronary artery bypass |
| Coronary artery disease |
| Decreased appetite |
| Deep vein thrombosis |
| Depression |
| — date |
| Diabetes mellitus |
| — date |
| Diverticulum |
| Drug hypersensitivity |
| Dyspepsia |
| Dyspnoea |
| Dysuria |
| Erectile dysfunction |
| Fatigue |
| — date |
| Gastritis |
| Gastrooesophageal reflux disease |
| Gout |
| Gynaecomastia |
| Haematuria |
| Haemorrhoids |
| Hernia repair |
| Hiatus hernia |
| Hot flush |
| — date |
| Hydronephrosis |
| Hypercholesterolaemia |
| — date |
| Hyperlipidaemia |
| Hypertension |
| — date |
| Hypothyroidism |
| Inguinal hernia |
| Insomnia |
| — date |
| Metastases to bone |
| — date |
| Musculoskeletal pain |
| Myocardial infarction |
| Myocardial ischaemia |
| Nausea |

Continued on next page

| Comorbitity |
|---|
| Nephrolithiasis |
| Nocturia |
| — date |
| Obesity |
| Oedema peripheral |
| Osteoarthritis |
| — date |
| Pain |
| Pain in extremity |
| Pollakiuria |
| — date |
| Renal cyst |
| Tobacco user |
| Tonsillectomy |
| Transient ischaemic attack |
| Urinary incontinence |
| Urinary retention |

### B.1.2 Medications

**Table B.2**: Extracted features corresponding to medications or therapies. Available for patients from all studies.

| Medication or Therapy |
|---|
| ACE inhibitors |
| Acetic acid derivatives and related substances |
| Adrenergics and oth.drugs for obstruct.airway dis. |
| Alpha adrenoreceptor antagonists |
| Alpha and beta blocking agents |
| Aminoalkyl ethers |
| Angiotensin II antagonists |
| Anilides |
| Anti androgens |
| — duration |
| Antibiotics |
| Antidepressants |
| Antihistamines |
| Antiinfectives |
| Antiinfl. prep. non steroids for topical use |
| Antiinflammatory agents non steroids |
| Antiinflammatory preparations non steroids for to |
| Antiinflammatory products for vaginal administrat. |
| Ascorbic acid vitamin C |
| Benzodiazepine related drugs |
| Benzothiazepine derivatives |
| Beta blocking agents |
| Biguanides |
| Bisphosphonates |

Continued on next page

| Medication or Therapy |
|---|
| Bulk producers |
| Calcium |
| Corticosteroids |
| Coxibs |
| Cutaneous |
| Digitalis glycosides |
| Dihydropyridine derivatives |
| Diphenylpropylamine derivatives |
| Diuretics |
| Electrolyte solutions |
| Estrogens |
| Fenamates |
| Fluoroquinolones |
| Folic acid and derivatives |
| Glucocorticoids |
| Gonadotropin releasing hormones |
| H2 receptor antagonists |
| Heparins |
| HMG COA reductase inhibitors |
| Hormonotherapy |
| — duration |
| Imidazole derivatives |
| Insulins and analogues |
| Iron trivalent |
| Laxatives |
| Lipid modifying agents |
| Magnesium |
| Multivitamins |
| Non selective monoamine reuptake inhibitors |
| Opium alkaloids |
| Oral |
| Organic nitrates |
| Other agents for local oral treatment |
| Other antianemic preparations |
| Other antiemetics |
| Other antiepileptics |
| Other antineoplastic agents |
| Other cardiac preparations |
| Other dermatologicals |
| Other ophthalmologicals |
| Other opioids |
| Other plain vitamin preparations |
| Other urologicals |
| Phenothiazines |
| Phenylpiperidine derivatives |
| Platelet aggregation inhibitors excl. heparin |
| Potassium |
| Preparations inhibiting uric acid production |
| Preparations with salicylic acid derivatives |
| Progestogens |

Continued on next page

| Medication or Therapy |
|---|
| Propionic acid derivatives |
| Propulsives |
| Proton pump inhibitors |
| Pyrazolones |
| Salicylic acid and derivatives |
| Selective beta 2 adrenoreceptor agonists |
| Selective serotonin reuptake inhibitors |
| Selenium |
| Serotonin antagonists |
| Softeners emollients |
| Sulfonamides |
| Testosterone 5 alpha reductase inhibitors |
| Thiazides |
| Thiazolidinediones |
| Thyroid hormones |
| Total chemical classes |
| Treatment for adverse event |
| Urinary antispasmodics |
| Vitamin B complex |
| Vitamin D and analogues |
| Vitamin K antagonists |
| Vitamins |

### B.1.3 Laboratory Measurements

**Table B.3**: Extracted features corresponding to laboratory measurements. •: Indicates that a particular feature (row) was used in a particular dataset (column).

| | MSKCC | Celgene | Sanofi | MSKCC<br>Celgene | MSKCC<br>Sanofi | Celgene<br>Sanofi | MSKCC<br>Celgene<br>Sanofi |
|---|---|---|---|---|---|---|---|
| Alanine transaminase | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |
| Albumin | | • | • | | | • | |
| — date | | • | • | | | • | |
| — range | | • | • | | | • | |
| Alkaline phosphatase | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |
| Aspartate aminotransferase | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |
| Basophils | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |

Continued on next page

| | MSKCC | Celgene | Sanofi | MSKCC<br>Celgene | MSKCC<br>Sanofi | Celgene<br>Sanofi | MSKCC<br>Celgene<br>Sanofi |
|---|---|---|---|---|---|---|---|
| Basophils-leukocytes ratio | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Calcium | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |
| Creatinine | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |
| Eosinophils | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Eosinophils-leukocytes ratio | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Hematocrit | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Hemoglobin | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |
| Lactate dehydrogenase | • | • | | • | | | |
| — date | • | • | | • | | | |
| — range | • | • | | • | | | |
| Lymphocytes | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Lymphocytes-leukocytes ratio | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Magnesium | | • | • | | | • | |
| — date | | • | • | | | • | |
| — range | | • | • | | | • | |
| Monocytes | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Monocytes-leukocytes ratio | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Neutrophils | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |
| Neutrophils-leukocytes ratio | | • | | | | | |

Continued on next page

| | MSKCC | Celgene | Sanofi | MSKCC | MSKCC | Celgene | MSKCC |
|---|---|---|---|---|---|---|---|
| | | | | Celgene | Sanofi | Sanofi | Celgene |
| | | | | | | | Sanofi |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Phosphorus | | • | • | | | • | |
| — date | | • | • | | | • | |
| — range | | • | • | | | • | |
| Platelet count | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |
| Potasium | | • | • | | | • | |
| — date | | • | • | | | • | |
| — range | | • | • | | | • | |
| Prostate specific antigen | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | | | • | | | | |
| Red blood cells | | • | | | | | |
| — date | | • | | | | | |
| — range | | • | | | | | |
| Sodium | | • | • | | | • | |
| — date | | • | • | | | • | |
| — range | | • | • | | | • | |
| Testosterone | • | | • | | • | | |
| — date | • | | • | | • | | |
| — range | • | | • | | • | | |
| Total bilirubin | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |
| Total protein | | • | • | | | • | |
| — date | | • | • | | | • | |
| — range | | • | • | | | • | |
| White blood cells | • | • | • | • | • | • | • |
| — date | • | • | • | • | • | • | • |
| — range | • | • | • | • | • | • | • |

### B.1.4 Tumor Measurements

**Table B.4**: Extracted features corresponding to tumor measurements. •: Indicates that a particular feature (row) was used in a particular dataset (column).

| | Celgene | Sanofi | Celgene |
|---|---|---|---|
| | | | Sanofi |
| # non-target lesions | • | • | • |
| # target lesions | • | • | • |
| Adrenal | | • | • |
| — date | | • | • |
| Bladder | • | • | • |

Continued on next page

| | Celgene | Sanofi | Celgene<br>Sanofi |
|---|---|---|---|
| Bone | ● | ● | ● |
| — date | ● | ● | ● |
| Liver | ● | ● | ● |
| — date | ● | ● | ● |
| Lungs | ● | ● | ● |
| — date | ● | ● | ● |
| Lymph nodes | ● | ● | ● |
| — nodes date | ● | ● | ● |
| Max lesion size | ● | ● | ● |
| Min lesion size | ● | ● | ● |
| Muscle or soft tissue | ● | ● | ● |
| Pleura | ● | ● | ● |
| Prostate | ● | ● | ● |
| – date | ● | ● | ● |

### B.1.5 Vital Signs

**Table B.5**: Extracted features corresponding to vital signs. ●: Indicates that a particular feature (row) was used in a particular dataset (column).

| | MSKCC | Celgene | Sanofi | MSKCC<br>Celgene | MSKCC<br>Sanofi | Celgene<br>Sanofi | MSKCC<br>Celgene<br>Sanofi |
|---|---|---|---|---|---|---|---|
| Date collected | ● | ● | ● | ● | ● | ● | ● |
| Diastolic blood pressure | | ● | ● | | | ● | |
| Pulse | | ● | | | | | |
| Systolic blood pressure | | ● | ● | | | ● | |

### B.1.6 Miscellaneous

**Table B.6**: List of features that were already available on a per patient level. ●: Indicates that a particular feature (row) was used in a particular dataset (column).

| | MSKCC | Celgene | Sanofi | MSKCC<br>Celgene | MSKCC<br>Sanofi | Celgene<br>Sanofi | MSKCC<br>Celgene<br>Sanofi |
|---|---|---|---|---|---|---|---|
| Abdominal lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| ACE inhibitors | ● | ● | ● | ● | ● | ● | ● |
| Adrenal lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Age | ● | ● | ● | ● | ● | ● | ● |
| Alanine transaminase | ● | ● | ● | ● | ● | ● | ● |
| Albumin | | ● | ● | | | ● | |
| Alkaline phosphatase | ● | ● | ● | ● | ● | ● | ● |
| Analgesics | ● | ● | ● | ● | ● | ● | ● |

Continued on next page

| | MSKCC | Celgene | Sanofi | MSKCC<br>Celgene | MSKCC<br>Sanofi | Celgene<br>Sanofi | MSKCC<br>Celgene<br>Sanofi |
|---|---|---|---|---|---|---|---|
| Anti-androgens | ● | ● | ● | ● | ● | ● | ● |
| Anti-estrogens | ● | ● | ● | ● | ● | ● | ● |
| Aspartate aminotransferase | ● | ● | ● | ● | ● | ● | ● |
| Beta blocking agents | ● | ● | ● | ● | ● | ● | ● |
| Bilateral lymphadenectomy | ● | ● | ● | ● | ● | ● | ● |
| Bilateral orchidectomy | ● | ● | ● | ● | ● | ● | ● |
| Bisphosponate | ● | ● | ● | ● | ● | ● | ● |
| Bladder lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Blood and lymphatic system | ● | ● | ● | ● | ● | ● | ● |
| Body mass index | ● | ● | ● | ● | ● | ● | ● |
| Bone lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Calcium | ● | ● | ● | ● | ● | ● | ● |
| Cardiac disorders | ● | ● | ● | ● | ● | ● | ● |
| Cerebrovascular accident | ● | ● | ● | ● | ● | ● | ● |
| Chronic obstructive pulmonary disease | ● | ● | ● | ● | ● | ● | ● |
| Congenital, familial and genetic | ● | ● | ● | ● | ● | ● | ● |
| Congestive heart failure | ● | ● | ● | ● | ● | ● | ● |
| Corticosteroid | ● | ● | ● | ● | ● | ● | ● |
| Creatinine | ● | ● | ● | ● | ● | ● | ● |
| Deep venous thrombosis | ● | ● | ● | ● | ● | ● | ● |
| Diabetes | ● | ● | ● | ● | ● | ● | ● |
| Ear and labyrinth | ● | ● | ● | ● | ● | ● | ● |
| ECOG performance status | ● | ● | ● | ● | ● | ● | ● |
| Endocrine disorders | ● | ● | ● | ● | ● | ● | ● |
| Estrogens | ● | ● | ● | ● | ● | ● | ● |
| Eye disorders | ● | ● | ● | ● | ● | ● | ● |
| Gastroesophageal reflux disease | ● | ● | ● | ● | ● | ● | ● |
| Gastrointestinal bleed | ● | ● | ● | ● | ● | ● | ● |
| Gastrointestinal disorders | ● | ● | ● | ● | ● | ● | ● |
| Gen disord and admin site | ● | ● | ● | ● | ● | ● | ● |
| Glucocorticoids | ● | ● | ● | ● | ● | ● | ● |
| Gomadotropin | ● | ● | ● | ● | ● | ● | ● |
| Height | ● | ● | ● | ● | ● | ● | ● |
| Hemoglobin | ● | ● | ● | ● | ● | ● | ● |
| Hepatobiliary disorders | ● | ● | ● | ● | ● | ● | ● |
| HMG COA reductase inhibitors | ● | ● | ● | ● | ● | ● | ● |
| Imidazole | ● | ● | ● | ● | ● | ● | ● |

Continued on next page

| | MSKCC | Celgene | Sanofi | MSKCC<br>Celgene | MSKCC<br>Sanofi | Celgene<br>Sanofi | MSKCC<br>Celgene<br>Sanofi |
|---|---|---|---|---|---|---|---|
| Immune system disorders | ● | ● | ● | ● | ● | ● | ● |
| Infections and infestations | ● | ● | ● | ● | ● | ● | ● |
| Injury, poison and procedural | ● | ● | ● | ● | ● | ● | ● |
| Investigations | ● | ● | ● | ● | ● | ● | ● |
| Kidney lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Lactate dehydrogenase | ● | ● | | ● | | | |
| Liver lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Lung lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Lymph node lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Lymphocytes | | ● | | | | | |
| Magnesium | | ● | ● | | | ● | |
| Metabolism and nutrition | ● | ● | ● | ● | ● | ● | ● |
| Musc/skeletal and connect tissue | ● | ● | ● | ● | ● | ● | ● |
| Myocardial infarction | ● | ● | ● | ● | ● | ● | ● |
| Neoplasms benign, malig and unspec | ● | ● | ● | ● | ● | ● | ● |
| Nervous system disorders | ● | ● | ● | ● | ● | ● | ● |
| Neutrophils | ● | ● | ● | ● | ● | ● | ● |
| Non-target lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Orchidectomy | ● | ● | ● | ● | ● | ● | ● |
| Other lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Pathological bone fractures | ● | ● | ● | ● | ● | ● | ● |
| Peptic ulcer disease | ● | ● | ● | ● | ● | ● | ● |
| Phosphorus | | ● | ● | | | ● | |
| Platelet count | ● | ● | ● | ● | ● | ● | ● |
| Pleura lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Prostate lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Prostate specific antigen | ● | ● | ● | ● | ● | ● | ● |
| Prostatectomy | ● | ● | ● | ● | ● | ● | ● |
| Psychiatric disorders | ● | ● | ● | ● | ● | ● | ● |
| Pulmonary embolism | ● | ● | ● | ● | ● | ● | ● |
| Race | ● | ● | ● | ● | ● | ● | ● |
| Radiotherapy | ● | ● | ● | ● | ● | ● | ● |
| Red blood cells | | ● | | | | | |
| Region of the world | | ● | ● | ● | ● | ● | ● |
| Renal and urinary disorders | ● | ● | ● | ● | ● | ● | ● |
| Resp, thoracic and mediastinal | ● | ● | ● | ● | ● | ● | ● |
| Skin and subcutaneous tissue | ● | ● | ● | ● | ● | ● | ● |
| Social circumstances | ● | ● | ● | ● | ● | ● | ● |

Continued on next page

| | MSKCC | Celgene | Sanofi | MSKCC<br>Celgene | MSKCC<br>Sanofi | Celgene<br>Sanofi | MSKCC<br>Celgene<br>Sanofi |
|---|---|---|---|---|---|---|---|
| Soft tissue lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Spinal cord compression | ● | ● | ● | ● | ● | ● | ● |
| Surgical and medical procedures | ● | ● | ● | ● | ● | ● | ● |
| Target lesion(s) | ● | ● | ● | ● | ● | ● | ● |
| Testosterone | | | ● | | | | |
| Total bilirubin | ● | ● | ● | ● | ● | ● | ● |
| Total protein | | ● | ● | | | ● | |
| Treatment | ● | ● | ● | ● | ● | ● | ● |
| Turp | ● | ● | ● | ● | ● | ● | ● |
| Vascular disorders | ● | ● | ● | ● | ● | ● | ● |
| Visceral metastases | ● | ● | ● | ● | ● | ● | ● |
| Weight | ● | ● | ● | ● | ● | ● | ● |
| White blood cells | ● | ● | ● | ● | ● | ● | ● |

## B.2 Hyper-Parameter Configurations

The list below provides a detailed list of hyper-parameter configurations used to build a heterogeneous ensemble of survival and regression models in the Prostate Cancer DREAM challenge (see chapter 7).

- Cox proportional hazards model with ridge penalty (13 configurations):
  - Penalty $\lambda$: $2^{-12}, 2^{-10}, \ldots, 2^{12}$
- Survival support vector machine (13 configurations):
  - Penalty $\gamma$: $2^{-12}, 2^{-10}, \ldots, 2^{12}$
- Random survival forest (24 configurations):
  - Number of of trees: 1,000
  - Minimum number of samples in a terminal node: 3, 5, 10, 25, 50, 100
  - Split criterion: log-rank splitting
  - Number of candidate splits to evaluate per feature: 2, 5, 10, $\infty$
- Gradient boosting with regression tree as base learner (1,728 configurations):
  - Number of iterations: 100, 500, 1000, 1500
  - Subsampling percentage: 100%, 75%, 50%
  - Learning rate: 0.06, 0.125, 0.25
  - Maximum number of leaf nodes: 5, 10, 20
  - Minimum number of samples per split: 2, 5, 10, 20
  - Maximum number of features to evaluate per split: all, $\sqrt{\text{\# features}}$, 50%, 75%
- Gradient boosting with componentwise least squares as base learner (36 configurations):
  - Number of iterations: 100, 500, 1000, 1500
  - Subsampling percentage: 100%, 75%, 50%

  - Learning rate: 0.06, 0.125, 0.25

# Bibliography

[1] A Medical Research Council Investigation, “Streptomycin treatment of pulmonary tuberculosis,” *BMJ*, vol. 2, no. 4582, pp. 769–782, 1948. DOI: 10.1136/bmj.2.4582.769.

[2] O. O. Aalen, “Nonparametric inference for a family of counting processes,” *The Annals of Statistics*, vol. 6, pp. 701–726, 1978. JSTOR: 2958850.

[3] Accenture. (2014). Insight driven health, Getting EMR back in the fast lane, [Online]. Available: https://www.accenture.com/us-en/~/media/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Industries_11/Accenture-Getting-EMR-Back-Fast-Lane.pdf (visited on 12/13/2015).

[4] A. Airola, T. Pahikkala, and T. Salakoski, “Training linear ranking SVMs in linearithmic time using red–black trees,” *Pattern Recognition Letters*, vol. 32, no. 9, pp. 1328–1336, 2011. DOI: 10.1016/j.patrec.2011.03.014.

[5] A. Aizerman, E. M. Braverman, and L. I. Rozoner, “Theoretical foundations of the potential function method in pattern recognition learning,” *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.

[6] H. Akaike, “Information theory and an extension of the maximum likelihood principle,” in *$2^{nd}$ International Symposium on Information Theory*, 1973, pp. 267–281. DOI: 10.1007/978-1-4612-1694-0_15.

[7] G. Ambler, S. Seaman, and R. Z. Omar, “An evaluation of penalised survival methods for developing prognostic models with rare events,” *Statistics in Medicine*, vol. 31, no. 11-12, pp. 1150–1161, 2012. DOI: 10.1002/sim.4371.

[8] P. K. Andersen, J. P. Klein, and M.-J. Zhang, “Testing for centre effects in multi-centre survival studies: A Monte Carlo comparison of fixed and random effects tests,” *Statistics in Medicine*, vol. 18, no. 12, pp. 1489–1500, 1999.

[9] F. J. Anscombe, “The transformation of Poisson, binomial and negative-binomial data,” *Biometrika*, vol. 35, no. 3-4, pp. 246–254, 1948. DOI: 10.1093/biomet/35.3-4.246.

[10] J. Ansell and M. Phillips, *Practical Methods for Reliability Data Analysis.* Clarendon Press, 1994.

[11] L. Antolini, P. Boracchi, and E. Biganzoli, “A time-dependent discrimination index for survival data,” *Statistics in Medicine*, vol. 24, no. 24, pp. 3927–3944, 2005. DOI: 10.1002/sim.2427.

[12] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008. DOI: 10.1007/s10994-007-5040-8.

[13] P. Armitage, "Trials and errors: The emergence of clinical statistics," *Journal of the Royal Statistical Society. Series A*, vol. 146, no. 4, pp. 321–334, 1983. JSTOR: 2981451.

[14] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, *et al.*, "Gene ontology: Tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000. DOI: 10.1038/75556.

[15] J. Ba and B. Frey, "Adaptive dropout for training deep neural networks," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 3084–3092.

[16] N. Bacaër, "Halley's life table," in *A Short History of Mathematical Population Dynamics*. Springer, 2011, ch. 2. DOI: 10.1007/978-0-85729-115-8.

[17] E. Bair, T. Hastie, D. Paul, and R. Tibshirani, "Prediction by supervised principal components," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 119–137, 2006. DOI: 10.1198/016214505000000628.

[18] C. Bartenhagen, H.-U. Klein, C. Ruckert, X. Jiang, and M. Dugas, "Comparative study of unsupervised dimension reduction techniques for the visualization of microarray gene expression data," *BMC Bioinformatics*, vol. 11, no. 1, p. 567, 2010. DOI: 10.1186/1471-2105-11-567.

[19] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009. DOI: 10.1137/080716542.

[20] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 14*, 2002, pp. 585–591. DOI: 10.1162/089976603321780317.

[21] R. Bender, T. Augustin, and M. Blettner, "Generating survival times to simulate Cox proportional hazards models," *Statistics in Medicine*, vol. 24, no. 11, pp. 1713–1723, 2005. DOI: 10.1002/sim.2059.

[22] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI," in *Large-Scale Kernel Machines*, L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds., MIT Press, 2007, pp. 321–360.

[23] Y. Bengio and M. Monperrus, "Non-local manifold tangent learning," in *Advances in Neural Information Processing Systems 17*, 2005, pp. 129–136.

[24] B. Benjamin and J. Graunt, "John Graunt's 'Observations'," *Journal of the Institute of Actuaries*, vol. 90, no. 1, pp. 1–61, 1964. JSTOR: 41139746.

[25] A. Benner, "Application of "aggregated classifiers" in survival time studies," in *Proc. in Computational Statistics: COMPSTAT*, W. Härdle and B. Rönz, Eds., 2002, pp. 171–176. DOI: 10.1007/978-3-642-57489-4_21.

[26] A. Benner, M. Zucknick, T. Hielscher, C. Ittrich, and U. Mansmann, "High-dimensional Cox models: The choice of penalty as part of the model building process," *Biometrical Journal*, vol. 52, no. 1, pp. 50–69, 2010. DOI: `10.1002/bimj.200900064`.

[27] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "Nearest Neighbor" meaningful?" In *Database Theory – ICDT'99*, vol. 1540, 1999, pp. 217–235. DOI: `10.1007/3-540-49257-7_15`.

[28] H. Binder and M. Schumacher, "Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models," *BMC Bioinformatics*, vol. 9, p. 14, 2008. DOI: `10.1186/1471-2105-9-14`.

[29] P. Blanche, J.-F. Dartigues, and H. Jacqmin-Gadda, "Review and comparison of ROC curve estimators for a time-dependent outcome with marker-dependent censoring," *Biometrical Journal*, vol. 55, no. 5, pp. 687–704, 2013. DOI: `10.1002/bimj.201200045`.

[30] C. I. Bliss and W. L. Stevens, "The calculation of the time-mortality curve," *Annals of Applied Biology*, vol. 24, no. 4, pp. 815–852, 1937. DOI: `10.1111/j.1744-7348.1937.tb05058.x`.

[31] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *5th Annual Workshop on Computational Learning Theory*, 1992. DOI: `10.1145/130385.130401`.

[32] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. DOI: `10.1561/2200000016`.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[34] M. K. Brawer, "Recent progress in the treatment of advanced prostate cancer with intermittent dose-intense calcitriol (DN-101)," *Reviews in Urology*, vol. 9, no. 1, pp. 1–8, 2007. [Online]. Available: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1831525/`.

[35] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 26, no. 2, pp. 123–140, 1996. DOI: `10.1023/A:1018054314350`.

[36] ——, "Using adaptive bagging to debias regressions," Department of Statistics, University of California, Berkeley, Tech. Rep., 1999.

[37] ——, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: `10.1023/a:1010933404324`.

[38] L. Breiman and A. Cutler. (Jun. 6, 2004). Random forests - classification description, [Online]. Available: `https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm` (visited on 11/02/2015).

[39] L. Breiman, J. H. Friedman, C. J. Stone, and R. A. Ohlsen, *Classification and Regression Trees*. Wadsworth International Group, 1984.

[40] N. Breslow, "Covariance analysis of survival data under the proportional hazards model," *International Statistics Review*, vol. 43, pp. 89–99, 1974. DOI: 10.2307/2529620.

[41] J. Buckley and I. James, "Linear regression with censored data," *Biometrika*, vol. 66, no. 3, pp. 429–436, 1979. DOI: 10.1093/biomet/66.3.429.

[42] P. Bühlmann, "Boosting for high-dimensional linear models," *The Annals of Statistics*, vol. 34, no. 2, pp. 559–583, 2006. DOI: 10.1214/009053606000000092.

[43] P. Bühlmann and T. Hothorn, "Boosting algorithms: Regularization, prediction and model fitting," *Statistical Science*, vol. 22, no. 4, pp. 477–505, 2007. DOI: 10.1214/07-STS242.

[44] P. Bühlmann and B. Yu, "Analyzing bagging," *The Annals of Statistics*, vol. 30, no. 4, pp. 927–961, 2002. JSTOR: 1558692.

[45] ——, "Boosting with the $L_2$ loss," *Journal of the American Statistical Association*, vol. 98, no. 462, pp. 324–339, 2003. DOI: 10.1198/016214503000125.

[46] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998. DOI: 10.1023/A:1009715923555.

[47] L. F. Burgette and J. P. Reiter, "Multiple imputation for missing data via sequential regression trees," *American Journal of Epidemiology*, vol. 172, no. 9, pp. 1070–1076, 2010. DOI: 10.1093/aje/kwq260.

[48] T. Cai, G. Tonini, and X. Lin, "Kernel machine approach to testing the significance of multiple genetic markers for risk prediction," *Biometrics*, vol. 67, no. 3, pp. 975–986, 2011. DOI: 10.1111/j.1541-0420.2010.01544.x.

[49] A. Carbonez, L. Györfi, and E. van der Meulen, "Partitioning-estimates of a regression function under random censoring," *Statistics & Decisions*, vol. 13, no. 1, pp. 21–37, 1995. DOI: 10.1524/strm.1995.13.1.21.

[50] R. Caruana, A. Munson, and A. Niculescu-Mizil, "Getting the most out of ensemble selection," in *6th IEEE International Conference on Data Mining*, 2006, pp. 828–833. DOI: 10.1109/icdm.2006.76.

[51] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *22nd International Conference on Machine Learning*, 2004. DOI: 10.1145/1015330.1015432.

[52] Centers for Medicare & Medicaid Services, "Medicare and medicaid programs; electronic health record incentive program-stage 3 and modifications to meaningful use in 2015 through 2017," *Federal Register*, no. 80 FR 62761, pp. 62 761–62 955, Oct. 16, 2015. [Online]. Available: https://federalregister.gov/a/2015-25595 (visited on 12/13/2015).

[53] O. Chapelle, "Training a support vector machine in the primal," *Neural Computation*, vol. 19, no. 5, pp. 1155–1178, 2007. DOI: 10.1162/neco.2007.19.5.1155.

[54] O. Chapelle and S. S. Keerthi, "Efficient algorithms for ranking with SVMs," *Information Retrieval*, vol. 13, no. 3, pp. 201–205, 2009. DOI: 10.1007/s10791-009-9109-9.

[55] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng, "Boosted multi-task learning," *Machine Learning*, vol. 85, no. 1-2, pp. 149–173, 2010. DOI: 10.1007/s10994-010-5231-6.

[56] H. J. Cho and S.-M. Hong, "Median regression tree for analysis of censored survival data," *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans*, vol. 38, no. 3, pp. 715–726, 2008. DOI: 10.1109/tsmca.2008.918598.

[57] A. Ciampi, S. A. Hogg, S. McKinney, and J. Thiffault, "RECPAM: A computer program for recursive partition and amalgamation for censored survival data and other situations frequently occurring in biostatistics. I. methods and program features," *Computer Methods and Programs in Biomedicine*, vol. 26, no. 3, pp. 239–256, 1988. DOI: 10.1016/0169-2607(88)90004-1.

[58] J. E. Ciecka, "Edmond Halley's life table and its uses," *Journal of Legal Economics*, vol. 15, no. 1, pp. 65–74, 2008.

[59] A. C. Cohen, *Truncated and Censored Samples, Theory and Applications.* CRC Press, 1991.

[60] J. Cohen, "A coefficient of agreement of nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960. DOI: 10.1177/001316446002000104.

[61] L. M. Collins, J. L. Schafer, and C.-M. Kam, "A comparison of inclusive and restrictive strategies in modern missing data procedures.," *Psychological Methods*, vol. 6, no. 4, pp. 330–351, 2001. DOI: 10.1037/1082-989x.6.4.330.

[62] M. S. Cookson, B. J. Roth, P. Dahm, C. Engstrom, S. J. Freedland, *et al.*, "Castration-resistant prostate cancer: AUA guideline," *The Journal of Urology*, vol. 190, no. 2, pp. 429–438, 2013. DOI: 10.1016/j.juro.2013.05.005.

[63] C. D. Correa and P. Lindstrom, "Locally-scaled spectral clustering using empty region graphs," in *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1330–1338. DOI: 10.1145/2339530.2339736.

[64] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. DOI: 10.1007/BF00994018.

[65] J. Costello, J. Guinney, L. Zhou, C. Bare, T. Wang, *et al.* (Jun. 30, 2015). Prostate cancer dream challenge, [Online]. Available: https://www.synapse.org/#!Synapse:syn2813558 (visited on 11/25/2015).

[66] R. Courant and D. Hilbert, *Methods of Mathematical Physics.* Wiley-Interscience, 1953.

[67] D. R. Cox, "Regression models and life tables," *Journal of the Royal Statistical Society: Series B*, vol. 34, pp. 187–220, 1972. JSTOR: 2985181.

[68] D. R. Cox and D. Oakes, "Accelerated life model," in *Analysis of Survival Data.* Chapman & Hall/CRC, 1984, pp. 64–70.

[69] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning," Microsoft Research, Tech. Rep. MSR-TR-2011-114, 2011.

[70] A. Daemen, D. Timmerman, T. Van den Bosch, C. Bottomley, E. Kirk, C. Van Holsbeke, L. Valentin, T. Bourne, and B. De Moor, "Improved modeling of clinical data with kernel methods," *Artificial Intelligence in Medicine*, vol. 54, pp. 103–114, 2012. DOI: `10.1016/j.artmed.2011.11.001`.

[71] R. B. Davis and J. R. Anderson, "Exponential survival trees," *Statistics in Medicine*, vol. 8, no. 8, pp. 947–961, 1989. DOI: `10.1002/sim.4780080806`.

[72] R. De Bin, W. Sauerbrei, and A.-L. Boulesteix, "Investigating the prediction ability of survival models based on both clinical and omics data: Two case studies," *Statistics in Medicine*, vol. 33, no. 30, pp. 5310–5329, 2014. DOI: `10.1002/sim.6246`.

[73] D. M. Delong, G. H. Guirguis, and Y. C. So, "Efficient computation of subset selection probablilities with application to Cox regression," *Biometrika*, vol. 81, no. 3, pp. 607–611, 1994. DOI: `10.1093/biomet/81.3.607`.

[74] R. S. Dembo and T. Steihaug, "Truncated newton algorithms for large-scale optimization," *Mathematical Programming*, vol. 26, no. 2, pp. 190–212, 1983.

[75] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[76] S. Derksen and H. J. Keselman, "Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables," *British Journal of Mathematical and Statistical Psychology*, vol. 45, no. 2, pp. 265–282, 1992. DOI: `10.1111/j.2044-8317.1992.tb00992.x`.

[77] C. Desmedt, F. Piette, S. Loi, Y. Wang, F. Lallemand, *et al.*, "Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the TRANSBIG multicenter independent validation series," *Clinical Cancer Research*, vol. 13, no. 11, pp. 3207–3214, 2007. DOI: `10.1158/1078-0432.CCR-06-2765`.

[78] T. G. Dietterich, "Ensemble methods in machine learning," in *1$^{st}$ International Workshop on Multiple Classifier Systems*, 2000, pp. 1–15. DOI: `10.1007/3-540-45014-9_1`.

[79] L. L. Doove, S. van Buuren, and E. Dusseldorp, "Recursive partitioning for missing data imputation in the presence of interaction effects," *Computational Statistics & Data Analysis*, vol. 72, pp. 92–104, 2014. DOI: `10.1016/j.csda.2013.10.025`.

[80] L. Duchateau and P. Janssen, *The Frailty Model.* Springer, 2008.

[81] B. Efron, "The efficiency of Cox's likelihood function for censored data," *Journal of the American Statistical Association*, vol. 72, no. 359, pp. 557–565, 1977. DOI: `10.1080/01621459.1977.10480613`.

[82] ——, "Estimating the error rate of a prediction rule: Improvement on cross-validation," *Journal of the American Statistical Association*, vol. 78, no. 382, pp. 316–331, 1983. DOI: `10.1080/01621459.1983.10477973`. JSTOR: `2288636`.

[83] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004. DOI: `10.1214/009053604000000067`.

[84] B. Efron and R. Tibshirani, "Improvements on cross-validation: The .632+ bootstrap method," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 548–560, 1997. DOI: 10.2307/2965703. JSTOR: 2965703.

[85] A. Eleuteri, "Support vector survival regression," in *4$^{th}$ IET International Conference on Advances in Medical, Signal and Information Processing*, 2008, pp. 1–4.

[86] A. Eleuteri and A. F. Taktak, "Support vector machines for survival regression," in *Computational Intelligence Methods for Bioinformatics and Biostatistics*, E. Biganzoli, A. Vellido, F. Ambrogi, and R. Tagliaferri, Eds., ser. LNCS, vol. 7548, Springer, 2012, pp. 176–189. DOI: 10.1007/978-3-642-35686-5.

[87] D. Engler and Y. Li, "Survival analysis with high-dimensional covariates: An application in microarray studies," *Statistical Applications in Genetics and Molecular Biology*, vol. 8, no. 1, pp. 1–22, 2009. DOI: 10.2202/1544-6115.1423.

[88] B. Epstein and M. Sobel, "Life testing," *Journal of the American Statistical Association*, vol. 48, no. 263, pp. 486–502, 1953. DOI: 10.1080/01621459.1953.10483488. JSTOR: 2281004.

[89] R. Etzioni, M. Pepe, G. Longton, C. Hu, and G. Goodman, "Incorporating the time dimension in receiver operating characteristic curves: A case study of prostate cancer," *Medical Decision Making*, vol. 19, no. 3, pp. 242–251, 1999. DOI: 10.1177/0272989x9901900303.

[90] L. Evers and C.-M. Messow, "Sparse kernel methods for high-dimensional survival data," *Bioinformatics*, vol. 24, no. 14, pp. 1632–1638, May 30, 2008. DOI: 10.1093/bioinformatics/btn253.

[91] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001. DOI: 10.1198/016214501753382273.

[92] W. Farr, "Influence of elevation on the fatality of cholera," *Journal of the Statistical Society of London*, vol. 15, no. 2, pp. 155–183, 1852. DOI: 10.2307/2338305. JSTOR: 2338305.

[93] ——, "Inequalities of the cholera mortality within the several parts of the water-fields," in *Report on the cholera epidemic of 1866 in England, Supplement to the twenty-ninth annual report of the Registrar-General.* 1868, pp. xxiv–xxv. [Online]. Available: http://www.histpop.org/ohpr/servlet/AssociatedPageBrowser?path=Browse&mno=690&pageseq=24&assoctitle=Cholera%20report,%201850 (visited on 12/11/2015).

[94] ——, *Vital statistics: A memorial volume of selections from the reports and writings of William Farr*, N. A. Humphreys, Ed. Offices of the Sanitary Institute, 1885. [Online]. Available: https://archive.org/details/vitalstatistics00humpgoog (visited on 12/11/2015).

[95] J. Ferlay, E. Steliarova-Foucher, J. Lortet-Tieulent, S. Rosso, J. Coebergh, H. Comber, D. Forman, and F. Bray, "Cancer incidence and mortality patterns in Europe: Estimates for 40 countries in 2012," *European Journal of Cancer*, vol. 49, no. 6, pp. 1374–1403, 2013. DOI: 10.1016/j.ejca.2012.12.027.

[96] R. A. Fisher, "On the interpretation of $\chi^2$ from contingency tables, and the calculation of P," *Journal of the Royal Statistical Society*, vol. 85, no. 1, pp. 87–94, 1922. JSTOR: 2340521.

[97] ——, "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London. Series A*, vol. 222, pp. 309–368, 1922. JSTOR: 91208.

[98] ——, *Statistical Methods for Research Workers.* Oliver & Boyd, 1925.

[99] ——, "Properties and applications of $H_h$ functions," *Mathematical Tables*, vol. 1, pp. 26–35, 1931.

[100] ——, *The Design of Experiments.* 1935.

[101] B. Fitzenberger and R. A. Wilke, "Using quantile regression for duration analysis," *Allgemeines Statistisches Archiv*, vol. 90, no. 1, pp. 105–120, 2006. DOI: 10.1007/s10182-006-0224-2.

[102] T. R. Fleming and D. P. Harrington, *Counting Processes and Survival Analysis.* John Wiley & Sons, Inc., 1991.

[103] D. Freedman, "A remark on the difference between sampling with and without replacement," *Journal of the American Statistical Association*, vol. 72, no. 359, p. 681, 1977. DOI: 10.1080/01621459.1977.10480637.

[104] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. JSTOR: 2699986.

[105] ——, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002. DOI: 10.1016/S0167-9473(01)00065-2.

[106] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976. DOI: 10.1016/0898-1221(76)90003-1.

[107] F. Galton, "An examination into the registered speeds of american trotting horses, with remarks on their value as hereditary data," *Proc. of the Royal Society of London*, vol. 62, pp. 310–315, 1897. JSTOR: 115734.

[108] E. A. Gehan, "A generalized Wilcoxon test for comparing arbitrarily singly-censored samples," *Biometrika*, vol. 52, no. 1-2, pp. 203–223, 1965. DOI: 10.1093/biomet/52.1-2.203.

[109] R. D. Gill, M. J. van der Laan, and J. M. Robins, "Coarsening at random: Characterizations, conjectures, counter-examples," in *1$^{st}$ Seattle Symposium in Biostatistics*, D. Y. Lin and T. R. Fleming, Eds., ser. Lecture Notes in Statistics, 1997, pp. 255–294. DOI: 10.1007/978-1-4684-6316-3_14.

[110] M. Glasser, "Exponential survival with covariance," *Journal of the American Statistical Association*, vol. 62, no. 318, pp. 561–568, 1967. DOI: 10.1080/01621459.1967.10482929. JSTOR: 2283983.

[111] D. Gleason, "Histologic grading and clinical staging of prostatic carcinoma," in *Urologic pathology: The prostate*, M. Tannenbaum, Ed. Philadelphia: Lea & Febiger, 1977, pp. 171–198.

[112] R. Glowinski and A. Marrocco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires," *Revue française d'automatique, informatique, recherche opérationnell*, vol. 9, no. 2, pp. 41–76, 1975.

[113] J. J. Goemann, "$L_1$ penalized estimation in the Cox proportional hazards model," *Biometrical Journal*, vol. 52, no. 1, pp. 70–84, 2010. DOI: `10.1002/bimj.200900028`.

[114] M. K. Gomel, B. Oldenburg, J. M. Simpson, M. Chilvers, and N. Owen, "Composite cardiovascular risk outcomes of a work-site intervention trial.," *American Journal of Public Health*, vol. 87, no. 4, pp. 673–676, 1997. DOI: `10.2105/ajph.87.4.673`.

[115] P. Good, *Permutation Tests, A Practical Guide to Resampling Methods for Testing Hypothesis*. Springer, 1994. DOI: `10.1007/978-1-4757-2346-5`.

[116] L. Gordon and R. A. Olshen, "Tree-structured survival analysis," *Cancer treatment reports*, vol. 69, pp. 1065–1069, 1985.

[117] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher, "Assessment and comparison of prognostic classification schemes for survival data," *Statistics in Medicine*, vol. 18, no. 17-18, pp. 2529–2545, 1999. DOI: `10.1002/(SICI)1097-0258(19990915/30)18:17/18<2529::AID-SIM274>3.0.CO;2-5`.

[118] J. Graunt, *Natural and Political Observations Mentioned in a following index, and made upon the Bills of Mortality*, 1st Edition. 1662. [Online]. Available: `http://www.edstephan.org/Graunt/bills.html`.

[119] J. Gui, D. Tao, Z. Sun, Y. Luo, X. You, and Y. Y. Tang, "Group sparse multiview patch alignment framework with view consistency for image classification," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 3126–3137, 2014. DOI: `10.1109/TIP.2014.2326001`.

[120] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[121] S. Halabi, C.-Y. Lin, W. K. Kelly, K. S. Fizazi, J. W. Moul, E. B. Kaplan, M. J. Morris, and E. J. Small, "Updated prognostic model for predicting overall survival in first-line chemotherapy for patients with metastatic castration-resistant prostate cancer," *Journal of Clinical Oncology*, vol. 32, no. 7, pp. 671–677, 2014. DOI: `10.1200/jco.2013.52.3696`.

[122] A. Hald, "Maximum likelihood estimation of the parameters of a normal distribution which is truncated at a known point," *Scandinavian Actuarial Journal*, vol. 1949, no. 1, pp. 119–134, 1949. DOI: `10.1080/03461238.1949.10419767`.

[123] E. Halley, "An estimate of the degrees of mortality of mankind, drawn from the curious tables of the births and funerals at the city of Breslaw, with an attempt to ascertain the price of annuities upon lives," *Philosophical Transactions*, vol. 17, pp. 596–610, 1693.

[124] Y. Han, F. Wu, D. Tao, J. Shao, Y. Zhuang, and J. Jiang, "Sparse unsupervised dimensionality reduction for multiple view data," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 10, pp. 1485–1496, 2012. DOI: `10.1109/TCSVT.2012.2202075`.

[125] J. A. Hanley and B. J. McNeil, “A method of comparing the areas under receiver operating characteristic curves derived from the same cases,” *Radiology*, vol. 148, no. 3, pp. 839–843, 1983.

[126] L. Hansen and P. Salamon, “Neural network ensembles,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990. DOI: 10.1109/34.58871.

[127] F. E. Harrell, R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati, “Evaluating the yield of medical tests,” *Journal of the American Medical Association*, vol. 247, no. 18, pp. 2543–2546, 1982. DOI: 10.1001/jama.1982.03320430047030.

[128] F. E. Harrell, K. L. Lee, and D. B. Mark, “Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors,” *Statistics in Medicine*, vol. 15, no. 4, pp. 361–387, 1996. DOI: 10.1002/(SICI)1097-0258(19960229)15:4<361::AID-SIM168>3.0.CO;2-4.

[129] T. Hastie, R. Tibshirani, and J. H. Friedman, “Boosting and additive trees,” in *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 1st Edition. 2001, ch. 10, pp. 299–345.

[130] ——, “Additive models, trees, and related methods,” in *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd Edition. 2013, ch. 9, pp. 295–336.

[131] ——, “Model assessment and selection,” in *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd Edition. 2013, ch. 7, pp. 219–260.

[132] ——, “Random forests,” in *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd Edition. 2013, ch. 15, pp. 587–604.

[133] X. He and P. Niyogi, “Locality preserving projections,” in *Advances in Neural Information Processing Systems 16*, 2004, pp. 153–160.

[134] P. J. Heagerty, T. Lumley, and M. S. Pepe, “Time-dependent ROC curves for censored survival data and a diagnostic marker,” *Biometrics*, vol. 56, no. 2, pp. 337–344, 2000. DOI: 10.1111/j.0006-341x.2000.00337.x.

[135] P. J. Heagerty and Y. Zheng, “Survival model predictive accuracy and ROC curves,” *Biometrics*, vol. 61, no. 1, pp. 92–105, 2005. DOI: 10.1111/j.0006-341X.2005.030814.x.

[136] D. F. Heitjan, “Annotation: What can be done about missing data? Approaches to imputation,” *American Journal of Public Health*, vol. 87, no. 4, pp. 548–550, 1997. DOI: 10.2105/ajph.87.4.548.

[137] D. F. Heitjan and D. B. Rubin, “Ignorability and coarse data,” *The Annals of Statistics*, vol. 19, no. 4, pp. 2244–2253, 1991. JSTOR: 2241929.

[138] R. Herbrich, T. Graepel, and K. Obermayer, “Large margin rank boundaries for ordinal regression,” in *Advances in Large Margin Classifiers*, A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. MIT Press, 2000, ch. 7, pp. 115–132.

[139] C. Heyde, "John Graunt," in *Statisticians of the Centuries*, C. Heyde, E. Seneta, P. Crépel, S. Fienberg, and J. Gani, Eds. Springer, 2001. DOI: 10.1007/978-1-4613-0179-0_3.

[140] G. Heywood, "Edmund Halley: Astronomer and actuary," *Journal of the Institute of Actuaries*, vol. 112, no. 2, pp. 278–301, 1985. JSTOR: 41140738.

[141] T. Hielscher, M. Zucknick, W. Werft, and A. Benner, "On the prognostic value of survival models with application to gene expression signatures," *Statistics in Medicine*, vol. 29, no. 7-8, pp. 818–829, 2010. DOI: 10.1002/sim.3768.

[142] A. B. Hill, *Principles of medical statistics.* 1937.

[143] C. Hinrichs, V. Singh, G. Xu, and S. C. Johnson, "Predictive markers for AD in a multi-modality framework: An analysis of MCI progression in the ADNI population," *NeuroImage*, vol. 55, no. 2, pp. 574–589, 2011. DOI: 10.1016/j.neuroimage.2010.10.081.

[144] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv: 1207.0580 [cs.NE].

[145] C. Hong, J. Yu, J. Li, and X. Chen, "Multi-view hypergraph learning by patch alignment framework," *Neurocomputing*, vol. 118, pp. 79–86, 2013. DOI: 10.1016/j.neucom.2013.02.017.

[146] D. Hosmer, S. Lemeshow, and S. May, *Applied Survival Analysis: Regression Modeling of Time to Event Data.* John Wiley & Sons, Inc., 2008.

[147] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933. DOI: 10.1037/h0071325.

[148] T. Hothorn, P. Bühlmann, S. Dudoit, A. Molinaro, and M. J. van der Laan, "Survival ensembles," *Biostatistics*, vol. 7, no. 3, pp. 355–373, 2006. DOI: 10.1093/biostatistics/kxj011.

[149] T. Hothorn and B. Lausen, "On the exact distribution of maximally selected rank statistics," *Computational Statistics & Data Analysis*, vol. 43, no. 2, pp. 121–137, 2003. DOI: 10.1016/S0167-9473(02)00225-6.

[150] C. Hou, C. Zhang, Y. Wu, and F. Nie, "Multiple view semi-supervised dimensionality reduction," *Pattern Recognotion*, vol. 43, no. 3, pp. 720–730, 2010. DOI: 10.1016/j.patcog.2009.07.015.

[151] C.-J. Hsiao and E. Hing, "Use and characteristics of electronic health record systems among office-based physician practices: United states, 2001–2013," *NCHS Data Brief*, no. 143, 2014. [Online]. Available: http://198.246.124.22/nchs/data/databriefs/db143.pdf (visited on 12/13/2015).

[152] J. Huang, S. Ma, and H. Xie, "Regularized estimation in the accelerated failure time model with high-dimensional covariates," *Biometrics*, vol. 62, no. 3, pp. 813–820, 2006. DOI: 10.1111/j.1541-0420.2006.00562.x.

[153] H. Hung and C. T. Chiang, “Estimation methods for time-dependent AUC models with survival data,” *Canadian Journal of Statistics*, vol. 38, no. 1, pp. 8–26, 2010. DOI: 10.1002/cjs.10046. JSTOR: 27805213.

[154] D. Hush, P. Kelly, C. Scovel, and I. Steinwart, “QP algorithms with guaranteed accuracy and run time for support vector machines,” *Journal of Machine Learning Research*, vol. 7, pp. 733–769, 2007.

[155] C. A. Hutchison, “DNA sequencing: Bench to bedside and beyond,” *Nucleic Acids Research*, vol. 35, no. 18, pp. 6227–6237, 2007. DOI: 10.1093/nar/gkm688.

[156] J. Hyde, “Survival analysis with incomplete observations,” in *Biostatistics Casebook*, R. G. Miller, B. Efron, B. W. Brown, and L. E. Moses, Eds. John Wiley & Sons, Inc., 1980, pp. 31–46.

[157] H. Ishwaran and U. B. Kogalur, “Random survival forests for R,” *R News*, vol. 7, no. 2, pp. 25–31, 2007. [Online]. Available: https://cran.r-project.org/doc/Rnews/Rnews_2007-2.pdf.

[158] H. Ishwaran, E. H. Blackstone, C. E. Pothier, and M. S. Lauer, “Relative risk forests for exercise heart rate recovery as a predictor of mortality,” *Journal of the American Statistical Association*, vol. 99, no. 467, pp. 591–600, 2004. DOI: 10.1198/016214504000000638.

[159] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer, “Random survival forests,” *The Annals of Applied Statistics*, vol. 2, no. 3, pp. 841–860, 2008. DOI: 10.1214/08-aoas169.

[160] H. Jeffreys, *The Theory of Probability.* Oxford University Press, 1961.

[161] P. B. Jensen, L. J. Jensen, and S. Brunak, “Mining electronic health records: Towards better research applications and clinical care,” *Nature Reviews. Genetics*, vol. 13, no. 6, pp. 395–405, 2012. DOI: 10.1038/nrg3208.

[162] H. Jin, L. Ying, K. Stone, and D. M. Black, “Alternative tree-structured survival analysis based on variance of survival time,” *Medical Decision Making*, vol. 24, no. 6, pp. 670–680, 2004. DOI: 10.1177/0272989x04271048.

[163] Z. Jin, D. Y. Lin, and Z. Ying, “On least-squares regression with censored data,” *Biometrika*, vol. 93, no. 1, pp. 147–161, 2006. DOI: 10.1093/biomet/93.1.147.

[164] B. A. Johnson, “On lasso for censored data,” *Electronic Journal of Statistics*, vol. 3, pp. 485–506, 2009. DOI: 10.1214/08-EJS322.

[165] J. D. Kalbfleisch and R. L. Prentice, *The Statistical Analysis of Failure Time Data.* John Wiley & Sons, Inc., 2002.

[166] W. B. Kannel, M. Feinleib, P. M. McNamara, R. J. Garrision, and W. P. Castelli, “An investigation of coronary heart disease in families: The Framingham Offspring Study,” *American Journal of Epidemiology*, vol. 110, no. 3, pp. 281–290, 1979.

[167] E. L. Kaplan and P. Meier, “Nonparametric estimation from incomplete observations,” *Journal of the American Statistical Association*, vol. 53, pp. 457–481, 1958. DOI: 10.2307/2281868. JSTOR: 2281868.

[168] I. Karlsson and J. Zhao, "Dimensionality reduction with random indexing: An application on adverse drug event detection using electronic health records," in *27th IEEE International Symposium on Computer-Based Medical Systems*, 2014, pp. 304–307. DOI: 10.1109/CBMS.2014.22.

[169] W. Karush, "Minima of functions of several variables with inequalities as side constraints," Master's thesis, Deptartment of Mathematics, University of Chicago, 1939.

[170] S. S. Keerthi and D. DeCoste, "A modified finite newton method for fast solution of large scale linear SVMs," *Journal of Machine Learning Research*, vol. 6, pp. 341–361, 2005.

[171] S. Keleş and M. R. Segal, "Residual-based tree-structured survival analysis," *Statistics in Medicine*, vol. 21, no. 2, pp. 313–326, 2002. DOI: 10.1002/sim.981.

[172] F. M. Khan and V. B. Zubek, "Support vector regression for censored data (SVRc): A novel tool for survival analysis," in *8th IEEE International Conference on Data Mining*, 2008, pp. 863–868. DOI: 10.1109/ICDM.2008.50.

[173] G. S. Kimeldorf and G. Wahba, "A correspondence between Bayesian estimation on stochastic processes and smoothing by splines," *The Annals of Mathematical Statistics*, vol. 41, no. 2, pp. 495–502, 1970. DOI: 10.1214/aoms/1177697089.

[174] M. Kirby, C. Hirst, and E. D. Crawford, "Characterising the castration-resistant prostate cancer population: A systematic review," *International Journal of Clinical Practice*, vol. 65, no. 11, pp. 1180–1192, 2011. DOI: 10.1111/j.1742-1241.2011.02799.x.

[175] J. Kittler, "Feature set search algorithms," in *Pattern recognition and signal processing.* Springer, 1978, pp. 41–60.

[176] J. P. Klein and M. L. Moeschberger, *Survival Analysis: Techniques for Censored and Truncated Data*, 2nd Edition. Springer, 2003.

[177] R. Koenker, *Quantile Regression.* Cambridge University Press, 2005.

[178] R. Koenker and G. J. Bassett, "Regression quantiles," *Econometrica*, vol. 46, no. 1, pp. 33–50, 1978. DOI: 10.2307/1913643.

[179] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *2nd Berkeley Symposium on Mathematical Statistics and Probabilistics*, Berkeley, 1951, pp. 481–492.

[180] T.-M. Kuo, C.-P. Lee, and C.-J. Lin, "Large-scale kernel RankSVM," in *SIAM International Conference on Data Mining*, 2014, pp. 812–820. DOI: 10.1137/1.9781611973440.93.

[181] T. D. Laajala, S. Khan, A. Airola, T. Mirtti, T. Pahikkala, P. Gopalacharyulu, and T. Aittokallio. (Nov. 3, 2015). Predicting patient survival and treatment discontinuation in DREAM 9.5 mCRPC challenge, [Online]. Available: https://www.synapse.org/#!Synapse:syn4227610/wiki/233734 (visited on 12/02/2015).

[182] S. Lagakos, L. Barraj, and V. De Gruttola, "Nonparametric analysis of truncated survival data, with application to AIDS," *Biometrika*, vol. 75, no. 3, pp. 515–523, 1988. DOI: 10.1093/biomet/75.3.515.

[183] J. Lambert and S. Chevret, "Summary measure of discrimination in survival models based on cumulative/dynamic time-dependent ROC curves," *Statistical Methods in Medical Research*, 2014. DOI: `10.1177/0962280213515571`.

[184] H. O. Lancaster, "Infectious diseases and microbiology," in *Quantitative methods in biological and medical sciences, A historical essay*. 1994, ch. 8.

[185] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[186] P. Langley, "The changing science of machine learning," *Machine Learning*, vol. 82, no. 3, pp. 275–279, 2011. DOI: `10.1007/s10994-011-5242-y`.

[187] C. P. Langlotz, "RadLex: A new method for indexing online educational materials," *Radiographics*, vol. 26, no. 6, pp. 1595–1597, 2006.

[188] M. LeBlanc and J. Crowley, "Relative risk trees for censored survival data," *Biometrics*, vol. 48, no. 2, pp. 411–425, 1992. DOI: `10.2307/2532300`.

[189] ——, "Survival trees by goodness of split," *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 457–467, 1993. DOI: `10.1080/01621459.1993.10476296`.

[190] C.-P. Lee and C.-J. Lin, "Large-scale linear RankSVM," *Neural Computation*, vol. 26, no. 4, pp. 781–817, 2014. DOI: `10.1162/NECO_a_00571`.

[191] H. Li and J. Gui, "Partial Cox regression analysis for high-dimensional microarray gene expression data," *Bioinformatics*, vol. 20 Suppl 1, pp. i208–15, 2004. DOI: `10.1093/bioinformatics/bth900`.

[192] H. Li and Y. Luan, "Kernel Cox regression models for linking gene expression profiles to censored survival data," in *Pacific Symposium on Biocomputing*, 2003, pp. 65–76.

[193] ——, "Boosting proportional hazards models using smoothing splines, with applications to high-dimensional microarray data," *Bioinformatics*, vol. 21, no. 10, pp. 2403–2409, 2005. DOI: `10.1093/bioinformatics/bti324`.

[194] Y. Li, B. Geng, D. Tao, Z.-J. Zha, L. Yang, and C. Xu, "Difficulty guided image retrieval using linear multiple feature embedding," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1618–1630, 2012. DOI: `10.1109/TMM.2012.2199292`.

[195] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, 2nd Edition. Wiley, 2002.

[196] S. Liu, L. Zhang, W. Cai, Y. Song, Z. Wang, L. Wen, and D. Feng, "A supervised multiview spectral embedding method for neuroimaging classification," in *20th IEEE International Conference on Image Processing*, 2013, pp. 601–605. DOI: `10.1109/ICIP.2013.6738124`.

[197] W.-Y. Loh and Y.-S. Shih, "Split selection methods for classification trees," *Statistica Sinica*, vol. 7, pp. 815–840, 1997.

[198] B. Long, P. S. Yu, and Z. Zhang, "A general model for multiple view unsupervised learning," in *SIAM International Conference on Data Mining*, 2008, pp. 822–833. DOI: `10.1137/1.9781611972788.74`.

[199] P. C. A. Louis, “Recherches sur les effets de la saignée dans plusieurs maladies inflammatoires,” *Archives Générales de Médecine*, vol. 18, pp. 321–336, 1828.

[200] ——, *Researches on the effects of bloodletting in some inflammatory diseases.* Hilliard, Gray & Company, 1936. [Online]. Available: `https://books.google.de/books?id=AlE5AQAAMAAJ&lr&pg=PA1` (visited on 12/11/2015).

[201] R. Luss and A. d’Aspremont, “Support vector machine classification with indefinite kernels,” in *Advances in Neural Information Processing Systems 20*, 2008, pp. 953–960.

[202] U. von Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007. DOI: `10.1007/s11222-007-9033-z`.

[203] D. Lynden-Bell, “A method for allowing for known observational selection in small samples applied to 3CR quasars,” *Monthly Notices of the Royal Astronomical Society*, vol. 155, pp. 95–118, 1971. DOI: `10.1093/mnras/155.1.95`.

[204] S. Ma, X. Song, and J. Huang, “Supervised group lasso with applications to microarray data analysis,” *BMC Bioinformatics*, vol. 8, no. 1, p. 60, 2007. DOI: `10.1186/1471-2105-8-60`.

[205] O. Mangasarian, “A finite newton method for classification,” *Optimization Methods and Software*, vol. 17, no. 5, pp. 913–929, 2002. DOI: `10.1080/1055678021000028375`.

[206] H. B. Mann and D. R. Whitney, “On a test of whether one of two random variables is stochastically larger than the other,” *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947. DOI: `10.1214/aoms/1177730491`.

[207] N. Mantel, “Evaluation of survival data and two new rank order statistics arising in its considerations,” *Cancer Chemotherapy Reports*, vol. 50, no. 3, pp. 163–170, 1966.

[208] N. Mantel and W. Haenszel, “Statistical aspects of the analysis of data from retrospective studies of disease,” *Journal of the National Cancer Institute*, vol. 22, no. 4, pp. 719–748, 1959. DOI: `10.1093/jnci/22.4.719`.

[209] T. G. Margineant and D. D. Dietterich, “Pruning adaptive boosting,” in *$14^{th}$ International Conference on Machine Learning*, 1997, pp. 211–218.

[210] T. Martinussen and L. Peng, “Alternatives to the Cox model,” in *Handbook of Survival Analysis*, J. P. Klein, H. C. van Houwelingen, J. G. Ibrahim, and T. H. Scheike, Eds. Chapman & Hall/CRC, 2014, pp. 49–75.

[211] A. Mayr and M. Schmid, “Boosting the concordance index for survival data – a unified framework to derive and evaluate biomarker combinations,” *PLoS One*, vol. 9, no. 1, e84483, 2014. DOI: `10.1371/journal.pone.0084483`.

[212] J. Mercer, “Functions of positive and negative type, and their connection with the theory of integral equations,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 209, no. 441-458, pp. 415–446, 1909. DOI: `10.1098/rsta.1909.0016`. JSTOR: `91043`.

[213] R. A. Moffitt, “New developments in econometric methods for labor market analysis,” in *Handbook of Labor Economics*, O. C. Ashenfelter and D. Card, Eds. Elsevier, 1999, vol. 3, Part A, ch. 24, pp. 1367–1397. DOI: `10.1016/S1573-4463(99)03005-9`.

[214] A. M. Molinaro, S. Dudoit, and M. J. van der Laan, “Tree-based multivariate regression and density estimation with right-censored data,” *Journal of Multivariate Analysis*, vol. 90, no. 1, pp. 154–177, 2004. DOI: 10.1016/j.jmva.2004.02.003.

[215] A. Morabia, “Observations made upon the bills of mortality,” *BMJ*, vol. 346, e8640, 2013. DOI: 10.1136/bmj.e8640.

[216] A. Morabia, “P. C. A. Louis and the birth of clinical epidemiology,” *Journal of Clinical Epidemiology*, vol. 49, no. 12, pp. 1327–1333, 1996. DOI: 10.1016/s0895-4356(96)00294-6.

[217] G. Ndrepepa, S. Braun, J. Mehilli, K. A. Birkmeier, R. A. Byrne, *et al.*, “Prognostic value of sensitive troponin T in patients with stable and unstable angina and undetectable conventional troponin.,” *American Heart Journal*, vol. 161, no. 1, pp. 68–75, 2011. DOI: 10.1016/j.ahj.2010.09.018.

[218] W. Nelson, “Theory and applications of hazard plotting for censored failure data,” *Technometrics*, vol. 14, pp. 945–965, 1972. DOI: 10.1080/00401706.1972.10488991.

[219] L. T. Newsome, R. S. Weller, J. C. Gerancher, M. A. Kutcher, and R. L. Royster, “Coronary artery stents: II. perioperative considerations and management,” *Anesthesia and Analgesia*, vol. 107, pp. 570–590, 2008. DOI: 10.1213/ane.0b013e3181732049.

[220] T. S. Nguyen and R. Javier, “Dimension reduction of microarray gene expression data: The accelerated failure time model,” *Journal of bioinformatics and computational biology*, vol. 7, no. 6, pp. 939–954, 2009. DOI: 10.1142/S0219720009004412.

[221] K. K. Nicodemus, “Letter to the editor: On the stability and ranking of predictors from random forest variable importance measures,” *Briefings in Bioinformatics*, vol. 12, no. 4, pp. 369–373, 2011. DOI: 10.1093/bib/bbr016.

[222] K. K. Nicodemus and J. D. Malley, “Predictor correlation impacts machine learning algorithms: Implications for genomic studies,” *Bioinformatics*, vol. 25, no. 15, pp. 1884–1890, 2009. DOI: 10.1093/bioinformatics/btp331.

[223] A. Omlin, C. Pezaro, and S. Gillessen Sommer, “Sequential use of novel therapeutics in advanced prostate cancer following docetaxel chemotherapy,” *Therapeutic Advances in Urology*, vol. 6, no. 1, pp. 3–14, 2014. DOI: 10.1177/1756287213509677.

[224] C. S. Ong, X. Mary, S. Canu, and A. J. Smola, “Learning with non-positive kernels,” in *$21^{st}$ International Conference on Machine learning*, 2004. DOI: 10.1145/1015330.1015443.

[225] C. Orsenigo and C. Vercellis, “A comparative study of nonlinear manifold learning methods for cancer microarray data classification,” *Expert Systems with Applications*, vol. 40, pp. 2189–2197, 2013. DOI: 10.1016/j.eswa.2012.10.044.

[226] Z. L. Óvári, “Kernels, eigenvalues and support vector machines,” PhD thesis, Australian National University, Canberra, 2000.

[227] M. Y. Park and T. Hastie, “$L_1$-regularization path algorithm for generalized linear models,” *Journal of the Royal Statistical Society: Series B*, vol. 69, no. 4, pp. 659–677, 2007. DOI: 10.1111/j.1467-9868.2007.00607.x.

[228] D. Paul, E. Bair, T. Hastie, and R. Tibshirani, "'Preconditioning' for feature selection and regression in high-dimensional problems," *The Annals of Statistics*, vol. 36, pp. 1595–1618, 2008. DOI: 10.1214/009053607000000578.

[229] K. Pearson, "On the systematic fitting of frequency curves," *Biometrika*, vol. 2, pp. 2–7, 1902.

[230] K. Pearson and A. Lee, "On the generalized probable error in multiple normal correlation," *Biometrika*, vol. 6, no. 1, pp. 59–68, 1908. DOI: 10.1093/biomet/6.1.59.

[231] T. E. Perry, H. Zha, K. Zhou, P. Frias, D. Zeng, and M. Braunstein, "Supervised embedding of textual predictors with applications in clinical diagnostics for pediatric cardiology," *Journal of the American Medical Informatics Association*, vol. 21, no. e1, e136–42, 2014. DOI: 10.1136/amiajnl-2013-01792.

[232] W. W. Peterson, T. G. Birdsall, and W. Fox, "The theory of signal detectability," *Transactions of the IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 171–212, 1954. DOI: 10.1109/TIT.1954.1057460.

[233] R. Peto and J. Peto, "Asymptotically efficient rank invariant procedures," *Journal of the Royal Statistical Society: Series A*, vol. 135, no. 2, pp. 185–207, 1972. DOI: 10.2307/2344317.

[234] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. MIT Press, 1999, pp. 185–208.

[235] S. Pölsterl, S. Conjeti, N. Navab, and A. Katouzian, "Survival analysis for high-dimensional, heterogeneous medical data: Exploring feature extraction as an alternative to feature selection," *Artificial Intelligence in Medicine*, 2016, submitted.

[236] S. Pölsterl, N. Navab, and A. Katouzian, "Fast training of support vector machines for survival analysis," in *Machine Learning and Knowledge Discovery in Databases*, A. Appice, P. P. Rodrigues, V. Santos Costa, J. Gama, A. Jorge, and C. Soares, Eds., ser. Lecture Notes in Computer Science, 2015, pp. 243–259. DOI: 10.1007/978-3-319-23525-7_15.

[237] C. Porzelius, M. Schumacher, and H. Binder, "Sparse regression techniques in low-dimensional survival data settings," *Statistics and Computing*, vol. 20, no. 2, pp. 151–163, 2010. DOI: 10.1007/s11222-009-9155-6.

[238] K. V. Rashmi and R. Gilad-Bachrach, "DART: Dropouts meet multiple additive regression trees," in *18$^{th}$ International Conference on Artificial Intelligence and Statistics*, 2015, pp. 489–497. arXiv: 1505.1866 [cs.LG].

[239] G. Ridgeway, "The state of boosting," *Computing Science and Statistics*, pp. 172–181, 1999.

[240] J. M. Robins and A. Rotnitzky, "Recovery of information and adjustment for dependent censoring using surrogate markers," in *AIDS Epidemiology*. 1992, pp. 297–331. DOI: 10.1007/978-1-4757-1229-2_14.

[241] N. Rooney, D. Patterson, S. Anand, and A. Tsymbal, "Dynamic integration of regression models," in *Proc. of the 5$^{th}$ International Workshop on Multiple Classifier Systems*, 2004, pp. 164–173. DOI: 10.1007/978-3-540-25966-4_16.

[242] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. DOI: 10.1037/h0042519.

[243] A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, *et al.*, "The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma," *New England Journal of Medicine*, vol. 346, no. 25, pp. 1937–1947, 2002. DOI: 10.1056/nejmoa012914.

[244] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000. DOI: 10.1126/science.290.5500.2323.

[245] S. T. Roweis, L. K. Saul, and G. E. Hinton, "Global coordination of local linear models," in *Advances in Neural Information Processing Systems 14*, 2002, pp. 889–896.

[246] D. B. Rubin, *Multiple imputation for nonresponse in surveys.* John Wiley & Sons, Inc., 1987.

[247] D. B. Rubin, "Multiple imputation after 18+ years," *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 473–489, 1996. DOI: 10.1080/01621459.1996.10476908. JSTOR: 2291635.

[248] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007. DOI: 10.1093/bioinformatics/btm344.

[249] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, pp. 210–229, 1959. DOI: 10.1147/rd.441.0206.

[250] G. A. Satten and S. Datta, "The Kaplan-Meier estimator as an inverse-probability-of-censoring weighted average," *The American Statistician*, vol. 55, no. 3, pp. 207–210, 2001. JSTOR: 2685801.

[251] J. L. Schafer, *Analysis of Incomplete Multivariate Data.* Chapman and Hall/CRC, 1997.

[252] J. L. Schafer and J. W. Graham, "Missing data: Our view of the state of the art," *Psychological Methods*, vol. 7, no. 2, pp. 147–177, 2002. DOI: 10.1037//1082-989X.7.2.147.

[253] M. Schmid and T. Hothorn, "Flexible boosting of accelerated failure time models," *BMC Bioinformatics*, vol. 9, p. 269, 2008. DOI: 10.1186/1471-2105-9-269.

[254] M. Schmid, H. A. Kestler, and S. Potapov, "On the validity of time-dependent AUC estimators," *Briefings in Bioinformatics*, vol. 16, no. 1, pp. 153–168, 2013. DOI: 10.1093/bib/bbt059.

[255] M. Schmid, M. Wright, and A. Ziegler. (2015). On the use of Harrell's C for node splitting in random survival forests. arXiv: 1507.03092 [stat.ML].

[256] J. Schofield, “In my lord the cardinal’s service,” in *The Rise and Fall of Thomas Cromwell, Henry VIII’s most faithful servant*. The History Press, 2011, ch. 1.

[257] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998. DOI: `10.1162/089976698300017467`.

[258] M. Schumacher, H. Binder, and T. Gerds, “Assessment of survival prediction models based on microarray data,” *Bioinformatics*, vol. 23, no. 14, pp. 1768–1774, 2007. DOI: `10.1093/bioinformatics/btm232`.

[259] G. Schwarz, “Estimating the dimension of a model,” *The Annals of Statistics*, vol. 6, pp. 461–464, 1978. DOI: `10.1214/aos/1176344136`.

[260] S. R. Seaman, J. W. Bartlett, and I. R. White, “Multiple imputation of missing covariates with non-linear effects and interactions: An evaluation of statistical methods,” *BMC Medical Research Methodology*, vol. 12, no. 1, p. 46, 2012. DOI: `10.1186/1471-2288-12-46`.

[261] M. R. Segal, “Regression trees for censored data,” *Biometrics*, vol. 44, no. 1, pp. 35–47, 1988. DOI: `10.2307/2531894`. JSTOR: `2531894`.

[262] A. D. Shah, J. W. Bartlett, J. Carpenter, O. Nicholas, and H. Hemingway, “Comparison of random forest and parametric imputation models for imputing missing data using MICE: A CALIBER study,” *American Journal of Epidemiology*, vol. 179, no. 6, pp. 764–774, 2014. DOI: `10.1093/aje/kwt312`.

[263] H. Shen, D. Tao, and D. Ma, “Multiview locally linear embedding for effective medical image retrieval,” *PLoS ONE*, vol. 8, no. 12, e82409, 2013. DOI: `10.1371/journal.pone.0082409`.

[264] P. K. Shivaswamy, W. Chu, and M. Jansche, “A support vector approach to censored targets,” in *7th IEEE International Conference on Data Mining*, 2007, pp. 655–660. DOI: `10.1109/ICDM.2007.93`.

[265] R. L. Siegel, K. D. Miller, and A. Jemal, “Cancer statistics, 2015,” *CA: A Cancer Journal for Clinicians*, vol. 65, no. 1, pp. 5–29, 2015. DOI: `10.3322/caac.21254`.

[266] H. A. Simon, “Why should machines learn?” In *Machine Learning, An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds. 1983, ch. 2, pp. 26–37. DOI: `10.1007/978-3-662-12405-5`.

[267] N. Simon, J. H. Friedman, T. Hastie, and R. Tibshirani, “Regularization paths for Cox’s proportional hazards model via coordinate descent,” *Journal of Statistical Software*, vol. 39, no. 5, pp. 1–13, 2011. DOI: `10.18637/jss.v039.i05`.

[268] E. H. Slate and B. W. Turnbull, “Statistical models for longitudinal biomarkers of disease onset,” *Statistics in Medicine*, vol. 19, no. 4, pp. 617–637, 2000. DOI: `10.1002/(sici)1097-0258(20000229)19:4<617::aid-sim360>3.0.co;2-r`.

[269] P. J. Smith, “Linear regression with censored data,” in *Analysis of Failure and Survival Data*. Chapman & Hall/CRC, 2002, pp. 187–214.

[270] A. J. Smola, Z. L. Óvári, and R. C. Williamson, “Regularization with dot-product kernels,” in *Advances in Neural Information Processing Systems 13*, 2001, pp. 308–314.

[271] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004. DOI: 10.1023/B:STCO.0000035301.49549.88.

[272] J. Snow, *On the Mode of Communication of Cholera*. 1855.

[273] S. J. Snow, "Commentary: Sutherland, Snow and water: The transmission of cholera in the nineteenth century," *International Journal of Epidemiology*, vol. 31, no. 5, pp. 908–911, 2002. DOI: 10.1093/ije/31.5.908.

[274] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[275] D. J. Stekhoven and P. Bühlmann, "MissForest – non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012. DOI: 10.1093/bioinformatics/btr597.

[276] W. L. Stevens, "The truncated normal distribution," *Annals of Applied Biology*, vol. 24, no. 4, pp. 815–852, 1937. DOI: 10.1111/j.1744-7348.1937.tb05058.x.

[277] J. Stewart, "Positive definite functions and generalizations, an historical survey," *Rocky Mountain Journal of Mathematics*, vol. 6, no. 3, pp. 409–434, 1976. DOI: 10.1216/RMJ-1976-6-3-409.

[278] N. J. Stone, J. G. Robinson, A. H. Lichtenstein, C. N. Bairey Merz, C. B. Blum, *et al.*, "2013 ACC/AHA guideline on the treatment of blood cholesterol to reduce atherosclerotic cardiovascular risk in adults," *Journal of the American College of Cardiology*, vol. 63, no. 25, pp. 2889–2934, 2014. DOI: 10.1016/j.jacc.2013.11.002.

[279] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, p. 307, 2008. DOI: 10.1186/1471-2105-9-307.

[280] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinformatics*, vol. 8, p. 25, 2007. DOI: 10.1186/1471-2105-8-25.

[281] W. Stute, "Consistent estimation under random censorship when covariables are present," *Journal of Multivariate Analysis*, vol. 45, no. 1, pp. 89–103, 1993. DOI: 10.1006/jmva.1993.1028.

[282] I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola, "Nonparametric quantile estimation," *Journal of Machine Learning Research*, vol. 7, pp. 1231–1264, 2006.

[283] I. F. Tannock, K. Fizazi, S. Ivanov, C. T. Karlsson, A. Fléchon, *et al.*, "Aflibercept versus placebo in combination with docetaxel and prednisone for treatment of men with metastatic castration-resistant prostate cancer (VENICE): A phase 3, double-blind randomised trial," *Lancet Oncology*, vol. 14, no. 8, pp. 760–768, 2013. DOI: 10.1016/S1470-2045(13)70184-0.

[284] R. E. Tarone and J. Ware, "On distribution-free tests for equality of survival distributions," *Biometrika*, vol. 64, no. 1, pp. 156–160, 1977. DOI: 10.1093/biomet/64.1.156.

[285] T. M. Therneau and E. J. Atkinson, "An introduction to recursive partitioning using the RPART routines," Mayo Foundation, Tech. Rep., Jun. 29, 2015. [Online]. Available: `https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf`.

[286] T. M. Therneau, P. M. Grambsch, and T. R. Fleming, "Martingale-based residuals for survival models," *Biometrika*, vol. 77, no. 1, pp. 147–160, 1990. DOI: `10.1093/biomet/77.1.147`.

[287] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996. JSTOR: `2346178`.

[288] ——, "The lasso method for variable selection in the Cox model," *Statistics in Medicine*, vol. 16, no. 4, pp. 385–395, 1997. DOI: `10.1002/(SICI)1097-0258(19970228)16:4<385::AID-SIM380>3.0.CO;2-3`.

[289] H. Tong, J. He, M. Li, C. Zhang, and W.-Y. Ma, "Graph based multi-modality learning," in *Proc. of the 13th ACM International Conference on Multimedia*, 2005, pp. 862–871. DOI: `10.1145/1101149.1101337`.

[290] G. Tutz and H. Binder, "Generalized additive modeling with implicit variable selection by likelihood-based boosting," *Biometrics*, vol. 62, no. 4, pp. 961–971, 2006. DOI: `10.1111/j.1541-0420.2006.00578.x`.

[291] ——, "Boosting ridge regression," *Computational Statistics & Data Analysis*, vol. 51, no. 12, pp. 6044–6059, 2007. DOI: `10.1016/j.csda.2006.11.041`.

[292] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L. J. Wei, "On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data," *Statistics in Medicine*, vol. 30, no. 10, pp. 1105–1117, 2011. DOI: `10.1002/sim.4154`.

[293] H. Uno, T. Cai, L. Tian, and L. J. Wei, "Evaluating prediction rules for t-year survivors with censored regression models," *Journal of the American Statistical Association*, vol. 102, pp. 527–537, 2007. DOI: `10.1198/016214507000000149`.

[294] V. Van Belle, K. Pelckmans, J. A. K. Suykens, and S. Van Huffel, "Support vector machines for survival analysis," in *Proc. of the 3rd International Conference on Computational Intelligence in Medicine and Healthcare*, 2007, pp. 1–8.

[295] ——, "Survival SVM: A practical scalable algorithm," in *16th European Symposium on Artificial Neural Networks*, 2008, pp. 89–94.

[296] V. Van Belle, K. Pelckmans, J. A. K. Suykens, and S. Van Huffel, "Learning transformation models for ranking and survival analysis," *Journal of Machine Learning Research*, vol. 12, pp. 819–862, 2011.

[297] V. Van Belle, K. Pelckmans, S. Van Huffel, and J. A. K. Suykens, "Support vector methods for survival analysis: A comparison between ranking and regression approaches," *Artificial Intelligence in Medicine*, vol. 53, no. 2, pp. 107–118, 2011. DOI: `10.1016/j.artmed.2011.06.006`.

[298] S. van Buuren, "Multiple imputation of discrete and continuous data by fully conditional specification," *Statistical Methods in Medical Research*, vol. 16, no. 3, pp. 219–242, 2007. DOI: `10.1177/0962280206074463`.

[299] S. van Buuren, J. P. Brand, C. G. Groothuis-Oudshoorn, and D. B. Rubin, “Fully conditional specification in multivariate imputation,” *Journal of Statistical Computation and Simulation*, vol. 76, no. 12, pp. 1049–1064, 2006. DOI: `10.1080/10629360600810434`.

[300] S. van Buuren and K. Oudshoorn, “Flexible multivariate imputation by MICE,” TNO Prevention and Health, Leiden, Tech. Rep. PG/VGZ/99.054, 1999.

[301] M. J. van der Laan and J. M. Robins, *Unified Methods for Censored Longitudinal Data and Causality.* Springer, 2003. DOI: `10.1007/978-0-387-21700-0`.

[302] L. J. P. Van Der Maaten, E. O. Postma, and H. J. Van Den Herik, “Dimensionality reduction: A comparative review,” *Journal of Machine Learning Research*, vol. 10, pp. 1–41, 2009. DOI: `10.1080/13506280444000102`.

[303] R. Van Noorden, B. Maher, and R. Nuzzo, “The top 100 papers, Nature explores the most-cited research of all time,” *Nature*, vol. 514, 7524 Oct. 29, 2014. [Online]. Available: `http://www.nature.com/news/the-top-100-papers-1.16224`.

[304] W. N. van Wieringen, D. Kun, R. Hampel, and A.-L. Boulesteix, “Survival prediction using gene expression data: A review and comparison,” *Computational Statistics & Data Analysis*, vol. 53, no. 5, pp. 1590–1603, 2009. DOI: `10.1016/j.csda.2008.05.021`.

[305] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer, 1995. DOI: `10.1007/978-1-4757-3264-1`.

[306] P. J. M. Verweij and H. C. Van Houwelingen, “Penalized likelihood in Cox regression,” *Statistics in Medicine*, vol. 13, no. 23-24, pp. 2427–2436, 1994. DOI: `10.1002/sim.4780132307`.

[307] G. Wahba, *Splines Models for Observational Data*, ser. Series in Applied Mathematics. SIAM, 1990, vol. 59.

[308] S. Wang, B. Nan, J. Zhu, and D. G. Beer, “Doubly penalized Buckley-James method for survival data with high-dimensional covariates,” *Biometrics*, vol. 64, no. 1, pp. 132–140, 2007. DOI: `10.1111/j.1541-0420.2007.00877.x`.

[309] X. Wang, W. Bian, and D. Tao, “Grassmannian regularized structured multi-view embedding for image classification,” *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2646–2660, 2013. DOI: `10.1109/TIP.2013.2255300`.

[310] Z. Wang and C. Wang, “Buckley-James boosting for survival analysis with high-dimensional biomarker data,” *Statistical Applications in Genetics and Molecular Biology*, vol. 9, no. 1, 2010. DOI: `10.2202/1544-6115.1550`.

[311] L. Wasserman, “Bayesian model selection and model averaging,” *Journal of Mathematical Psychology*, vol. 44, no. 1, pp. 92–107, 2000. DOI: `10.1006/jmps.1999.1278`.

[312] W. Weibull, “A statistical theory of the strength of materials,” *Ingeniors Vetenskaps Akakemien Handlingar 151*, pp. 293–297, 1939.

[313] ——, “Statistical distribution of wide applicability,” *Journal of Applied Mechanics*, vol. 18, pp. 293–297, 1951.

[314] E. A. Weiss, “Biographies: Eloge: Arthur Lee Samuel (1901-90),” *IEEE Annals of the History of Computing*, vol. 14, no. 3, pp. 55–69, 1992. DOI: `10.1109/85.150082`.

[315] I. R. White and P. Royston, “Imputing missing covariate values for the Cox model,” *Statistics in Medicine*, vol. 28, no. 15, pp. 1982–1998, 2009. DOI: `10.1002/sim.3618`.

[316] I. R. White, P. Royston, and A. M. Wood, “Multiple imputation using chained equations: Issues and guidance for practice,” *Statistics in Medicine*, vol. 30, no. 4, pp. 377–399, 2011. DOI: `10.1002/sim.4067`.

[317] F. Wilcoxon, “Individual comparison by ranking methods,” *Biometrics*, vol. 1, pp. 80–83, 1945. JSTOR: `3001968`.

[318] S. Windecker, P. Kolh, F. Alfonso, J.-P. Collet, J. Cremer, *et al.*, “2014 ESC/EACTS guidelines on myocardial revascularization,” *European Heart Journal*, vol. 35, no. 37, pp. 2541–2619, 2014.

[319] D. H. Wolpert, “The lack of a priori distinctions between learning algorithms,” *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, 1996. DOI: `10.1162/neco.1996.8.7.1341`.

[320] D. H. Wolpert and W. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997. DOI: `10.1109/4235.585893`.

[321] J. N. Wu, K. M. Fish, C. P. Evans, R. W. deVere White, and M. A. Dall’Era, “No improvement noted in overall or cause-specific survival for men presenting with metastatic prostate cancer over a 20-year period,” *Cancer*, vol. 120, no. 6, pp. 818–823, 2013. DOI: `10.1002/cncr.28485`.

[322] T. Xia, D. Tao, T. Mei, and Y. Zhang, “Multiview spectral embedding,” *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, vol. 40, no. 6, pp. 1438–1446, 2010. DOI: `10.1109/TSMCB.2009.2039566`.

[323] B. Xie, Y. Mu, D. Tao, and K. Huang, “m-SNE: Multiview stochastic neighbor embedding,” *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, vol. 41, no. 4, pp. 1088–1096, 2011. DOI: `10.1109/TSMCB.2011.2106208`.

[324] L. Yang, J. Liu, X. Yang, and X.-S. Hua, “Multi-modality web video categorization,” in *Proc. of the International Workshop on Multimedia Information Retrieval*, 2007, pp. 265–274. DOI: `10.1145/1290082.1290119`.

[325] H. Yu, J. Kim, Y. Kim, S. Hwang, and Y. H. Lee, “An efficient method for learning nonlinear ranking SVM functions,” *Information Sciences*, vol. 209, pp. 37–48, 2012. DOI: `10.1016/j.ins.2012.03.022`.

[326] K. Yu, Z. Wang, M. Hagenbuchner, and D. D. Feng, “Spectral embedding based facial expression recognition with multiple features,” *Neurocomputing*, vol. 129, pp. 136–145, 2014. DOI: `10.1016/j.neucom.2013.09.046`.

[327] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B*, vol. 68, no. 1, pp. 49–67, 2006. DOI: `10.1111/j.1467-9868.2005.00532.x`.

[328] H. H. Zhang and W. Lu, “Adaptive lasso for Cox’s proportional hazards model,” *Biometrika*, vol. 94, no. 3, pp. 691–703, 2007. DOI: `10.1093/biomet/asm037`.

[329] H. Zhang, "Splitting criteria in survival trees," in *10th International Workshop on Statistical Modeling*, G. U. H. Seeber, B. J. Francis, R. Hatzinger, and G. Steckel-Berger, Eds., 1995, pp. 305–313. DOI: `10.1007/978-1-4612-0789-4_37`.

[330] W. Zhang, T. Ota, V. Shridhar, J. Chien, B. Wu, and R. Kuang, "Network-based survival analysis reveals subnetwork signatures for predicting outcomes of ovarian cancer treatment," *PLoS Computational Biology*, vol. 9, no. 3, e1002975, 2013. DOI: `10.1371/journal.pcbi.1002975`.

[331] C. Zippin and P. Armitage, "Use of concomitant variables and incomplete survival information in the estimation of an exponential survival parameter," *Biometrics*, vol. 22, no. 4, pp. 665–672, 1966. DOI: `10.2307/2528067`. JSTOR: `2528067`.

[332] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B*, vol. 67, no. 2, pp. 301–320, 2005. DOI: `10.1111/j.1467-9868.2005.00503.x`.

# Author's Publications

[M1] S. Pölsterl, S. Conjeti, N. Navab, and A. Katouzian, "Survival analysis for high-dimensional, heterogeneous medical data: Exploring feature extraction as an alternative to feature selection," *Artificial Intelligence in Medicine*, 2016, submitted.

[M2] S. Pölsterl, N. Navab, and A. Katouzian, "Fast training of support vector machines for survival analysis," in *Machine Learning and Knowledge Discovery in Databases*, A. Appice, P. P. Rodrigues, V. Santos Costa, J. Gama, A. Jorge, and C. Soares, Eds., ser. Lecture Notes in Computer Science, 2015, pp. 243–259. DOI: 10.1007/978-3-319-23525-7_15.

[M3] S. Pölsterl, M. Singh, A. Katouzian, N. Navab, A. Kastrati, L. Ladic, and A. Kamen, "Stratification of coronary artery disease patients for revascularization procedure based on estimating adverse effects," *BMC Medical Informatics and Decision Making*, vol. 15, no. 1, p. 9, 2015. DOI: 10.1186/s12911-015-0131-0.

[M4] F. Graf, H.-P. Kriegel, S. Pölsterl, M. Schubert, and A. Cavallaro, "Position prediction in CT volume scans," in *Proc. of the 28th International Conference on Machine Learning (ICML). Workshop on Learning for Global Challenges*, 2011.

[M5] F. Graf, H.-P. Kriegel, M. Schubert, S. Pölsterl, and A. Cavallaro, "2D image registration in CT images using radial image descriptors," *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 607–614, 2011. DOI: 10.1007/978-3-642-23629-7_74.

[M6] J. Krumsiek, S. Pölsterl, D. M. Wittmann, and F. J. Theis, "Odefy – from discrete to continuous models," *BMC Bioinformatics*, vol. 11, no. 1, p. 233, 2010. DOI: 10.1186/1471-2105-11-233.