

TECHNISCHE UNIVERSITÄT MÜNCHEN  
Lehrstuhl für Mensch-Maschine-Kommunikation

# Immersive Interactive Data Mining and Machine Learning Algorithms for Big Data Visualization

Mohammadreza Babae

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktor-Ingenieurs (Dr.-Ing.)**

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. sc.techn. Andreas Herkersdorf

Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. habil. Gerhard Rigoll  
2. Univ.-Prof. Dr.-Ing. habil. Dirk Wollherr  
3. Prof. Dr. Mihai Datcu

Die Dissertation wurde am 13.08.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 16.02.2016 angenommen.



---

# Acknowledgments

This thesis could not have been written without the help, support, and advice of many people. First of all, I would like to thank my first examiner and PhD supervisor, Prof. Gerhard Rigoll, for providing me with the opportunity to work at the Institute for Human-Machine Communication at TUM and also for the advice and support he gave me during my PhD project. Moreover, I would like to acknowledge and extend my heartfelt gratitude to Prof. Mihai Datcu as the research group leader at Munich Aerospace. His expertise, guidance, and support kept me motivated during this project. I would like to thank the staff of Munich Aerospace, specially Dr. Veronika Gumpinger and Dr. Eva Rogowicz-Grimm who provided me with financial support to participate in scientific events and conferences. Additionally, I would like to thank all scientific and non-scientific members of the Institute for Human-Machine Communication. Here, I would like to mention Dr. Michael Dorr, Dr. Jürgen Geiger, Dr. Martin Hofmann, Dr. Florian Laquai, Dr. Felix Weninger, Nicolas Lehment, Erik Marchi, Daniel Merget, Simon Schenk, Philipp Tiefenbacher, and Yue Zhang. Moreover, I would like to thank Peter Brand, Marion Bächle, Heiner Hundhammer, and Martina Römp.

Finally, I want to thank my wife, parents and sisters for all their continuous love and support during all stages of my life. I especially want to thank God, who made all things possible.

Munich, May 17, 2016

Mohammadreza Babae



---

# Abstract

The amount of collected Earth Observation (EO) images is increasing exponentially and their growth is currently in the order of several terabytes per day. Therefore, the ability to automatically store and retrieve these images based on their content is highly desired. Traditional approaches are not accurate and robust enough to handle this massive amount of data. However, the combination of artificial intelligence and human intelligence could deliver promising results. Therefore, this thesis addresses several challenges in the field of human-machine communication for data mining applications. This is mainly done by first introducing an Immersive Visual Data Mining (IVDM) system, including image collections and feature space visualizations, interactive dimensionality reduction, and active learning for image classification. A Cave Automatic Virtual Environment (CAVE) is employed to support the user-image interactions and also immersive data visualization, which allows the user to navigate through the images and explore them. The feature space is visualized by applying state-of-the-art dimensionality reduction techniques to reduce the dimensionality to 3D. Additionally, a novel algorithm based on Non-negative Matrix Factorization (NMF) is developed to arrange the images in 3D space by decreasing the occlusion among images and to make use of the display space more efficiently. Two interactive dimensionality reduction algorithms are introduced to enhance the discriminative property of the features by incorporating the user-image interactions. To annotate images, a novel active learning algorithm is proposed to choose the most informative images for labeling. Finally, experimental evaluations using publicly available data sets demonstrate the efficiency of the proposed algorithms.



---

# Zusammenfassung

Die Anzahl aufgenommener Erdoberobservationsbilder (EO) steigt exponentiell mit einem aktuellen Wachstum in der Größenordnung von mehreren Terabyte pro Tag. Es ist daher notwendig, diese Bilder automatisch nach ihren Inhalten klassifizieren und durchsuchen zu können. Traditionelle Ansätze sind unpräzise oder störanfällig bei solch großen Datenmengen. Ein vielversprechender Ansatz hingegen ist die Klassifikation mittels künstlicher Intelligenz, welche zusätzlich durch menschliche Intelligenz unterstützt wird. Diese Dissertation adressiert mehrere Herausforderungen im Feld der Mensch-Maschine-Kommunikation für Anwendungen im Bereich Data Mining. Dies wird erreicht, indem zunächst ein Immersives Visuelles Data Mining (IVDM) System vorgestellt wird, welches die interaktive Darstellung von Bildern entsprechend ihrer hochdimensionalen Merkmalsrepräsentation ermöglicht. Mithilfe aktiven Lernens wird es ermöglicht, in dieser virtuellen Umgebung eine interaktive Dimensionsreduktion durchzuführen. Zur Förderung der Nutzerinteraktion und -immersion werden die Daten in einer Cave Automatic Virtual Environment (CAVE) dargestellt. So ist eine Navigation und Erkundung der Bilder in 3D möglich. Der Merkmalsraum wird durch die Anwendung von aktuellen Algorithmen zur Dimensionsreduktion auf 3D eingegrenzt. Zusätzlich wird ein neuer Algorithmus basierend auf Nichtnegativer Matrixfaktorisierung (NMF) entwickelt, welcher Überlappungen von Bildern durch Repositionierung in der 3D-Darstellung reduziert und den darstellbaren Raum effizienter ausnutzt. Weiterhin werden zwei interaktive Algorithmen zur Dimensionsreduktion vorgestellt, welche die Nutzerinteraktionen ausnutzen, um eine bessere Repräsentation im Merkmalsraum zu erreichen. Um die Bilder zu annotieren, wird ein neuer aktiver Lernalgorithmus präsentiert, welcher die informativsten Bilder für das Labeling automatisch auswählt. Zuletzt wird die Effektivität dieser Algorithmen durch experimentelle Evaluierungen auf öffentlich verfügbaren Datenbasen demonstriert.





---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Immersive visual data mining . . . . .	2
1.2	Structure of this thesis . . . . .	3
<b>2</b>	<b>Discriminative Data Representation</b>	<b>7</b>
2.1	Related work . . . . .	8
2.2	A review of NMF . . . . .	9
2.3	Discriminative NMF . . . . .	11
2.3.1	Optimization . . . . .	11
2.3.2	Computational complexity . . . . .	13
2.3.3	Experiment 1 . . . . .	13
2.3.4	Experiment 2 . . . . .	14
2.3.5	Convergence study . . . . .	17
2.3.6	Parameter analysis . . . . .	21
2.3.7	Locality preservation . . . . .	22
2.4	Attributes constrained NMF . . . . .	23
2.4.1	A Review of Relative Attributes . . . . .	23
2.4.2	Experiments . . . . .	26
2.5	Attributes constrained dictionary learning . . . . .	28
2.5.1	Proposed method . . . . .	31
2.5.2	Experiments . . . . .	34
2.6	Summary and conclusion . . . . .	38
<b>3</b>	<b>Immersive Visualization of Image Collections and Feature Space</b>	<b>41</b>
3.1	The Cave Automatic Virtual Environment . . . . .	42
3.1.1	The CAVE's components . . . . .	42
3.2	Data visualization . . . . .	45
3.2.1	Neighborhood graph and tree . . . . .	46

3.2.2	Feature space . . . . .	46
3.3	Assessment of DR using communication channel model . . . . .	47
3.3.1	Communication channel model . . . . .	49
3.3.2	Experiments . . . . .	50
3.4	A customized dimensionality reduction . . . . .	52
3.4.1	Regularized NMF for visualization . . . . .	54
3.4.2	Experiments . . . . .	58
3.5	Summary and conclusion . . . . .	67
<b>4</b>	<b>Interactive Dimensionality Reduction</b>	<b>69</b>
4.1	Related work . . . . .	69
4.2	Immersive data visualization . . . . .	71
4.3	Interactive algorithms . . . . .	72
4.3.1	Variance constrained NMF . . . . .	73
4.3.2	Center Map NMF . . . . .	75
4.3.3	Immersive interface . . . . .	76
4.3.4	Experiment 1 . . . . .	76
4.4	Pair-wise constrained NMF . . . . .	83
4.4.1	Experiment 2 . . . . .	85
4.5	Set-wise constrained NMF . . . . .	86
4.6	Summary and conclusions . . . . .	90
<b>5</b>	<b>Active Learning</b>	<b>91</b>
5.1	A review of active learning . . . . .	92
5.1.1	Membership query synthesis . . . . .	92
5.1.2	Stream-based selective sampling . . . . .	93
5.1.3	Pool-based active learning . . . . .	94
5.1.4	Training models . . . . .	99
5.2	State-of-the-art algorithms . . . . .	104
5.2.1	Transductive experimental design . . . . .	104
5.2.2	Locally linear reconstruction . . . . .	106
5.2.3	Manifold adaptive experimental design . . . . .	107
5.2.4	Support vector machine active learning . . . . .	109
5.3	Proposed method . . . . .	110
5.3.1	Trace-norm regularized Classifier (TC) . . . . .	110
5.3.2	Active learning with TC . . . . .	116
5.3.3	Visualization-based sample selection . . . . .	116
5.4	Experiments . . . . .	120
5.4.1	Data sets . . . . .	120
5.4.2	Setup . . . . .	121
5.4.3	Design 1: Active learning using TC . . . . .	121

5.4.4 Design 2: Visualization-based active learning . . . . .	125
5.5 Summary and conclusion . . . . .	128
<b>6 Summary and Conclusion</b>	<b>129</b>
6.1 Summary . . . . .	129
6.2 Conclusion and outlook . . . . .	130
<b>A Data sets</b>	<b>133</b>
A.1 Synthetic Aperture Radar (SAR) . . . . .	133
A.2 Caltech10 . . . . .	134
A.3 UC Merced Land Use . . . . .	134
A.4 Corel . . . . .	135
A.5 CMU PIE Faces . . . . .	136
A.6 AT&T ORL Faces . . . . .	136
A.7 Yale Faces . . . . .	137
A.8 Handwritten Digits . . . . .	138
<b>B Convergence Proofs</b>	<b>139</b>
B.1 Convergence Proof of DNMF . . . . .	139
B.2 Convergence Proof of VISNMF . . . . .	142
B.3 Convergence Proof of CMNMF . . . . .	145
B.4 Convergence Proof of VNMF . . . . .	147
B.5 Convergence Proof of Pairwise-NMF . . . . .	149
<b>Acronyms</b>	<b>153</b>
<b>List of Symbols</b>	<b>155</b>
<b>List of Figures</b>	<b>162</b>
<b>List of Tables</b>	<b>163</b>
<b>References</b>	<b>181</b>
<b>Publications by Author</b>	<b>184</b>
<b>Supervised Students' Theses</b>	<b>185</b>



# Introduction

Advances in sensing, communication, and storage devices have led to an exponential growth in the amount and types of data such as multimedia (e.g, image, video, text), Earth Observation (EO), social networks, astronomy, and scientific and engineering data. Therefore, searching, analyzing, exploring, and visualizing this massive amount of data, so-called *Big Data*, has become a gigantic challenge. Accordingly, the development of novel intelligent methods for exploring and understanding the contents of the available data is highly necessary. Over the past decade, many machine-learning-based systems have been devised to automatically explore the contents of the available data. These systems usually require large amounts of annotated data in their training phase. Moreover, they are normally treated as a black box in which the human has minimal intervention. Therefore, the gap between human understanding and machine understanding of the data content, the so-called *semantic gap*, gives rise to low performance. One promising solution is bridging this gap by the combination of machine intelligence with the human intelligence. However, the main challenge here is how to involve human intelligence in the learning process of machine, which is actually considered to be one of the main challenges in the area of human-machine communication.

Over the last two decades, various interactive learning scenarios have been introduced to include the human in learning process. Perhaps, the simplest human-machine interaction system is Query by Example (QE). In this system, the user inputs example data, and the machine delivers similar data samples to the user's query. However, the obtained results may not be satisfactory. In order to allow the user to inform the machine about the relevance of the provided results, the Relevance Feedback (RF) [Rui+98] system was proposed. This system improves the results by receiving binary (i.e, relevant or irrelevant) or graded relevancy feedback from the user in an iterative way. In a more complex system known as Active Learning (AL) [Set10], annotation and learning is performed at the same time. In AL, the user first annotates (labels) some samples of the data to train the learning algorithm. Using the learned model, the machine then labels the unlabeled data. Thereafter,

the obtained annotations are verified by the user and this process is run iteratively to annotate all of the entire unlabeled data. One previous study in the AL domain is [Jin+06] which allows the user to browse an image collection according to the semantic content of the images. This system shows the images in 2D on a computer screen based on Multidimensional Scaling (MDS) [TC10], a dimensionality reduction method. Then, users are provided with a set of interactions such as search by sample images and content relationship detection. Although this tool can provide the users with some intuitions about the annotation results, it does not assist them in understanding the global structure behind the contents of the entire image collection.

In every interactive learning system, both machine and human should communicate effectively via an interface. First, an interactive interface should be developed, in which the output of the machine is visualized for the human and on the other hand, the human is able to interact with both machine and data. Second, proper interactive learning algorithms should be developed such that the human is allowed to influence their performance. In this thesis, we describe an immersive visual data mining system that includes immersive visualization of data and interactive learning.

### 1.1 Immersive visual data mining

The proposed immersive visual data mining system, whose diagram is depicted in Figure 1.1, includes three processing blocks and an immersive interactive visualization tool [Bab+13a; Bab+13c; Bab+13b]. The system is built based on the intersection of Virtual Reality (VR) and Machine Learning (ML). Virtual reality technology is used to build an immersive interactive visualization tool that is responsible for visualizing the data and capturing the user interactions. The proposed machine learning algorithms are implemented as processing blocks embedded in the system. These algorithms are able to include the user's interactions in their learning process. The input of the system is an image repository with corresponding features, where the primitive feature extraction has been accomplished offline. The processing blocks are:

- **the Visualization of Image Collections** block aims to visualize the images by (1) applying modern dimensionality reduction to the corresponding feature vectors to determine the position of images in 3D, and (2) a constrained Non-negative Matrix Factorization (NMF) algorithm that controls the similarity and occlusion among images. The system allows users to manipulate the structure of the feature space based on their current understanding of the data. The manipulation is done by selecting, weighting, moving, and annotating the feature points or images. For example, the user can move the points to be closer or farther to the others based on their contents' similarities. These interactions in addition to the trajectory of the user in the feature space provide the system

with highly informative feedback, which is used to set the parameters (e.g., switching between the feature descriptors and the dimensionality reduction techniques).

- **the Interactive Dimensionality Reduction** block utilizes the user’s interactions to (1) reduce the dimensionality of features, and (2) increase the discriminative property of new features. There are two novel algorithms based on NMF incorporating the user’s interactions as constraints in the factorization process. Basically, a small fraction of images are visualized as a set of clusters in the Cave Automatic Virtual Environment (CAVE) and the user interacts with them by moving one image from one cluster to another. These interactions are fed into the factorization process to generate (learn) new features.
- **the Active Learning for Image Annotation and Classification** block aims to annotate the images and train the classifier simultaneously by (1) visualizing the output of a training model (i.e., classifier), and (2) selecting the most informative images for annotation. A novel and simple sample selection algorithm for annotation is introduced that outperforms several state-of-the art algorithms. There are two classifiers, namely Support Vector Machines (SVM) and Trace-norm regularized Classifier (TC), which are used as training models.

## 1.2 Structure of this thesis

This thesis presents the contributions of the proposed visual data mining system. Each aforementioned processing block is covered thoroughly in a separate chapter. Moreover, one extra chapter (i.e., Chapter 2) has been included to introduce and discuss a novel discriminative NMF algorithm. The chapters are as follows:

- Chapter 2 first presents a review of NMF and its related work. Then it introduces a discriminative NMF algorithm [Bab+ara]. It starts with a review of NMF and then describes how the label information of part of the data could enhance the discriminative property of features. In addition, the required analysis of the algorithm, including computational complexity, convergence plots, and experimental validation carried out on several publicly available data sets, are provided. In the end, we explain how relative attributes, as semantic information, can be used in the proposed algorithm for dimensionality reduction and also in dictionary learning.
- Chapter 3 describes first the technology behind the CAVE, including hardware, visualization software, and external libraries and tools [BRD13a]. Then, the visualization of feature spaces and image collections in 3D in two different ways are discussed. First, the images are positioned in 3D space based on

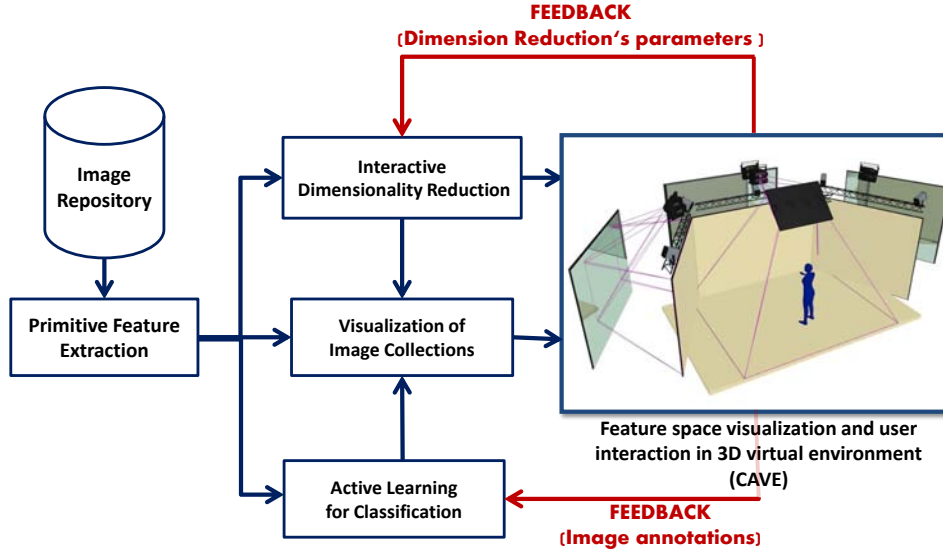


Figure 1.1: The diagram of the proposed visual data mining system. The contents of image repository are represented by feature vectors and fed into three processing blocks, namely interactive dimensionality reduction, visualization, and active learning. All these three blocks are connected with the CAVE. The learning algorithms incorporate the user’s feedback from the cave in the learning process and send the results again into the CAVE (i.e., human in the loop).

applying modern dimensionality reduction techniques to their corresponding feature vectors. However, this approach could lead to high occlusion among images, and therefore the visibility of the images decreases. To address this problem, a second approach is proposed to develop a constrained NMF, aiming to minimize the occlusion and use the display space more efficiently, while keeping the structure of the feature space unchanged [Bab+14a; Bab+arb].

- Chapter 4 talks about the interactive dimensionality reduction processing block. Here, two novel interactive dimensionality reduction algorithms are introduced. The user-image interactions in the CAVE are incorporated in the form of a regularizer coupled with the main objective function of NMF. The main properties and performance of these algorithms are covered at the end of the chapter.



- Chapter 5 presents the details of the active learning processing block. It first elaborates on a recently introduced classifier, the TC, and compares it with SVM in an active learning scenario [Bab+15a]. Then, a novel active learning algorithm is introduced that outperforms the state-of-the-art algorithms [Bab+15b; Bab+14b].
- Chapter 6 provides a summary of the thesis followed by a conclusion and suggestions for future work.



# Discriminative Data Representation

Today, we are dealing with the problem of processing large amounts of data, like the tremendous amount of satellite images, the huge number of texture and video files in databases, and uncountable bits of information on the Internet. In most cases, a matrix is used to store and represent the content of each data sample, in which each row (column) represents the content of one data point. For instance, a gray level image and a color image can be represented by three-dimensional and two-dimensional matrices, respectively. However, providing the storage space needed to store the data as well as the computational power necessary to process high-dimensional matrices pose a big challenge. Under such circumstances, matrix factorization and dictionary learning are attracting a lot of attention and play a vital role in Big Data processing. The content of this chapter is derived from our article to appear in the *Elsevier Journal of Neurocomputing* [Bab+ara].

In this chapter, we propose Discriminative Non-negative Matrix Factorization (DNMF) algorithms in order to generate (learn) discriminative features from original ones. The main contributions of the chapter are:

- A new label-constrained Non-negative Matrix Factorization (NMF) is introduced by coupling a discriminative regularizer to the main objective function of NMF.
- The updating rules for the factorization variables to obtain the optimal values and the convergence proof are obtained.
- The locality preserving property of the algorithm is studied and compared to a locality preserving NMF method. Additionally, the projection of synthetic data is compared with another discriminative NMF technique.
- The usage of relative attributes in matrix factorization is discussed.

- A new Discriminative Dictionary Learning (DDL) algorithm that uses relative attributes as semantic information.

Section 2.1 introduces state-of-the-art matrix factorization techniques with special focus on related works in the area of NMF. Section 2.2 briefly provides the background of NMF and related works in semi-supervised NMF. Section 2.3 presents the details of DNMF, followed by its computational complexity and the proof of convergence. The main difference between DNMF and other semi-supervised NMF methods is described at the end of this section. We describe the experiments performed on synthetic and real datasets in Section 2.3.4. We start by introducing our datasets and the evaluation metrics of the clustering process. Then, we study the convergence rate of the proposed algorithm. Additionally, the locality preserving property of DNMF is studied at the end of this section. Finally, in Section 2.6, a summary of the chapter is presented.

### 2.1 Related work

Matrix factorization techniques are quite often used in data analysis, storage, and visualization due to their ability to extract the most useful representation from the data content [He+05; MS07; GX11; Jol05]. Perhaps the most well-known and most widely used matrix factorization techniques are Principal Component Analysis (PCA) [Jol05], Singular Value Decomposition (SVD) [DHS12], and Non-negative Matrix Factorization [LS99]. The goal of these methods is to provide a compact low-dimensional representation of original data for further processes such as learning and visualization.

NMF itself is an unsupervised learning algorithm that decomposes a non-negative data matrix into two (or three) non-negative matrices, one of which is considered as a new representation of original data [WZ13; LS99]. This factorization leads to a parts-based representation of data, which is widely used in different applications such as face recognition [LS99], clustering [Liu+13; EF13; XLG03], hyperspectral unmixing [GP13; JQ09], music enhancement [LCS13], sparse coding [Vol+14], and graph matching [Jia+14]. Since the invention of NMF, many variants of this algorithm have been proposed to obtain a customized representation of data. For example, Graph Regularized Non-negative Matrix Factorization (GNMF) [Cai+11] preserves the locality property of data by utilizing the Laplacian of the neighborhood graph in its regularization term. Dual Graph Regularized Non-negative Matrix Factorization (DGNMF) [SJW12] and Multiple Graph Regularized Non-negative Matrix Factorization (MGNMF) [WBG13] are other variants of GNMF that include more constraints on the main objective function. Liu et al. [Hai+12] present a semi-supervised NMF approach, namely Constrained Non-negative Matrix Factorization (CNMF), utilizing the label information of part of the data embedded in the main objective function of NMF. In this algorithm, the data points with the same label are presented as a

single data point in the new feature space. As another example, Subspace Learning via Locally Constrained A-optimal Non-negative Projection (LCA), a semi-supervised local-structure preserving algorithm, is proposed. This uses the regularization term of GNMF in CNMF to simultaneously enhance the locality preserving and discriminative properties of new features. Nevertheless, the points with the same label are represented as a single point [Li+13].

The main disadvantage of previous semi-supervised NMF (i.e., CNMF and LCA) methods is that the data points decrease in number as available label information increases. The proposed DNMF utilizes the label information in a regularization term. The key idea of this approach is that the data points belonging to the same class should be very close together or aligned on the same axis in the new representation, but should not merge into a single point (like in CNMF or LCA). This regularizer increases the discriminative property of data points, which is controlled by only a single parameter. However, the range of this parameter, which delivers the best result, remains constant for different data sets, as the experiments confirm.

## 2.2 A review of NMF

Non-negative matrix factorization is a relatively novel framework for dimensionality reduction and data representation. It mainly incorporates the non-negativity constraint in the factorization process and therefore obtains a parts-based representation. In most cases, the observations are stored in data matrices or tensors. We assume that the content of an image repository is represented by a data matrix  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ , where  $\mathbf{x}_i$  is the representation (feature vector) of the  $i$ th data sample,  $N$  is the number of samples, and  $D$  is the dimension of the feature vectors. With a new reduced dimension  $K$ , the NMF algorithm approximates the matrix  $\mathbf{X}$  by a product of two non-negative matrices  $\mathbf{U} \in \mathbb{R}^{D \times K}$  and  $\mathbf{V} \in \mathbb{R}^{N \times K}$ :

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T. \quad (2.1)$$

$\mathbf{U}$  can be considered as the set of basis vectors and  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_N]^T$  as the coordinates of each sample with respect to these basis vectors. Therefore, the matrix  $\mathbf{V}$  is treated as the new feature vectors (or data representation). There are several cost functions that are able to measure the quality of this approximation. The two most popular cost functions are (1) the square of the Frobenius norm of the matrix differences, and (2) the Kullback-Leibler Divergence (KLD) of the two matrices [LS01]. For the Frobenius norm, the cost function is defined as

$$\begin{aligned} \min O_F &= \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 \\ \text{s.t. } \mathbf{U} &= [u_{ik}] \geq 0 \\ \mathbf{V} &= [v_{jk}] \geq 0, \end{aligned} \quad (2.2)$$

and for the KLD, the cost function is defined as

$$\begin{aligned} \min O_D &= \sum_{i,j} \left( x_{ij} \log \frac{x_{ij}}{\sum_k u_{ik} v_{jk}} - x_{ij} + \sum_k u_{ik} v_{jk} \right) \\ \text{s.t. } \mathbf{U} &= [u_{ik}] \geq 0 \\ \mathbf{V} &= [v_{jk}] \geq 0. \end{aligned} \quad (2.3)$$

Although both functions are convex with respect to  $\mathbf{U}$  and  $\mathbf{V}$  separately, they are not convex in both variables together. Therefore, many local minima exist. Lee et al. [LS01] present the update rules for minimizing these two cost functions and prove their convergence. For the Frobenius cost function (2.2), the update rules are

$$u_{ik} \leftarrow u_{ik} \frac{(\mathbf{XV})_{ik}}{(\mathbf{UV}^T\mathbf{V})_{ik}}, \quad v_{jk} \leftarrow v_{jk} \frac{(\mathbf{X}^T\mathbf{U})_{jk}}{(\mathbf{VU}^T\mathbf{U})_{jk}}, \quad (2.4)$$

and for the divergence cost function (2.3), the update rules are

$$u_{ik} \leftarrow u_{ik} \frac{\sum_j (x_{ij} v_{jk} / \sum_k u_{ik} v_{jk})}{\sum_j v_{jk}}, \quad v_{jk} \leftarrow v_{jk} \frac{\sum_i (x_{ij} u_{ik} / \sum_k u_{ik} v_{jk})}{\sum_i u_{ik}}. \quad (2.5)$$

Next, we consider the case of a semi-supervised setting. Without loss of generality, we assume that out of the samples  $\mathbf{X}$ , label information is available for the first  $N_l$  samples and there exist  $S$  classes in total.

Liu et al. [Hai+12] propose a semi-supervised NMF algorithm, namely CNMF. They assume that  $S$  classes exist and the label of the first  $N_l$  data points are available. They introduce a matrix  $\mathbf{C} \in \mathbb{R}^{N_l \times S}$  with  $c_{i,j} = 1$  if  $\mathbf{x}_i$  is labeled with class  $j$  and  $c_{i,j} = 0$  otherwise. Based on the matrix  $\mathbf{C}$ , matrix  $\mathbf{A}$  is defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{C} & 0 \\ 0 & \mathbf{I}_{N-N_l} \end{bmatrix}, \quad (2.6)$$

where  $\mathbf{I}$  is an  $(N - N_l) \times (N - N_l)$  identity matrix. Then, the matrix  $\mathbf{V}$  of samples in the new representation is expressed with the help of matrix  $\mathbf{A}$  and an auxiliary matrix  $\mathbf{Z}$ , where  $\mathbf{V} = \mathbf{AZ}$ . This means that if samples  $i$  and  $j$  have the same label, then  $\mathbf{v}_i = \mathbf{v}_j$ . With the help of the introduced matrices,  $\mathbf{X}$  is now approximated as  $\mathbf{X} \approx \mathbf{U}(\mathbf{AZ})^T$ . This is achieved, as before, by defining a cost function and minimizing over the variables  $\mathbf{U}$  and  $\mathbf{Z}$  [Hai+12]. Although CNMF has good performance, the samples with the same label merge into a single sample in the new representation, which may not be desirable in some applications such as visualization. Another semi-supervised NMF method [Li+13], which is actually a combination of GNMF and CNMF, uses the same trick to incorporate the label information and therefore has the same disadvantage.

## 2.3 Discriminative NMF

A discriminative NMF algorithm should factorize a data matrix such that the data points of the same class are separable from other data points in the new feature space. We assume that the label information is stored in matrix  $\mathbf{Q} \in \mathbb{R}^{S \times N}$  as

$$q_{ij} = \begin{cases} 1 & \text{if sample } j \text{ is labeled and belongs to class } i \\ 0 & \text{otherwise,} \end{cases}$$

where  $S$  is the number of classes (categories) and  $N$  is the total number of data points. For instance, consider the case of  $N = 8$  samples, out of which  $N_l = 5$  are labeled with the sample categories  $c_1 = 1, c_2 = 2, c_3 = 1, c_4 = 3, c_5 = 2$ . In this case, the matrix  $\mathbf{Q}$  would look like

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Having introduced the matrix  $\mathbf{Q}$ , a regularizer is introduced to be coupled with the Frobenius-NMF objective

$$O_L = \alpha \|\mathbf{Q} - \mathbf{A}\mathbf{V}_l^T\|^2, \quad (2.7)$$

with  $\mathbf{V}_l = [\mathbf{v}_1, \dots, \mathbf{v}_{N_l}, \mathbf{0}, \dots, \mathbf{0}]^T \in \mathbb{R}^{N \times K}$  and the matrix  $\mathbf{A} \in \mathbb{R}^{S \times K}$ , which linearly transforms and scales the vectors in the new representation in order to obtain the best fit for the matrix  $\mathbf{Q}$ . The matrix  $\mathbf{A}$  is allowed to take negative values and is computed as part of the NMF minimization. This regularizer can be considered as a linear regression term based on the labeled samples. Therefore, we approach the following constrained optimization problem:

$$\begin{aligned} \min O &= \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 + \alpha \|\mathbf{Q} - \mathbf{A}\mathbf{V}_l^T\|^2 \\ \text{s.t. } \mathbf{U} &= [u_{ik}] \geq 0 \\ \mathbf{V} &= [v_{jk}] \geq 0. \end{aligned} \quad (2.8)$$

Since the elements of  $\mathbf{A}\mathbf{V}_l^T$  might be negative, the KLD cost function is not applicable here.

### 2.3.1 Optimization

To optimize (2.8) and obtain the update rules for  $\mathbf{U}$  and  $\mathbf{V}$ , we first expand the objective function to

$$\begin{aligned} O &= \text{Tr}(\mathbf{X}\mathbf{X}^T) - 2\text{Tr}(\mathbf{X}\mathbf{V}\mathbf{U}^T) + \text{Tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) \\ &\quad + \alpha \text{Tr}(\mathbf{Q}\mathbf{Q}^T) - \alpha 2\text{Tr}(\mathbf{Q}\mathbf{V}_l\mathbf{A}^T) + \alpha \text{Tr}(\mathbf{A}\mathbf{V}_l^T\mathbf{V}_l\mathbf{A}^T). \end{aligned} \quad (2.9)$$

where  $\text{Tr}$  is Trace operator. We introduce Lagrange multipliers  $\Phi = [\phi_{ik}]$  and  $\Psi = [\psi_{jk}]$  for the constraints  $[u_{ik}] \geq 0$  and  $[v_{jk}] \geq 0$ , respectively. By adding these Lagrange multipliers to (2.9) and ignoring the constant terms, we come up with the Lagrangian:

$$\begin{aligned} \mathcal{L} = & -2\text{Tr}(\mathbf{X}\mathbf{V}\mathbf{U}^T) + \text{Tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) + \text{Tr}(\Phi\mathbf{U}) + \text{Tr}(\Psi\mathbf{V}) \\ & - \alpha 2\text{Tr}(\mathbf{Q}\mathbf{V}_l\mathbf{A}^T) + \alpha\text{Tr}(\mathbf{A}\mathbf{V}_l^T\mathbf{V}_l\mathbf{A}^T). \end{aligned} \quad (2.10)$$

The partial derivatives of  $\mathcal{L}$  with respect to  $\mathbf{U}$ ,  $\mathbf{V}$  and  $\mathbf{A}$  are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = -2\mathbf{X}\mathbf{V} + 2\mathbf{U}\mathbf{V}^T\mathbf{V} + \Phi \quad (2.11)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = -2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U} - \alpha 2\mathbf{Q}^T\mathbf{A} + \alpha 2\mathbf{V}_l\mathbf{A}^T\mathbf{A} + \Psi \quad (2.12)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}} = -2\mathbf{Q}\mathbf{V}_l + 2\mathbf{A}\mathbf{V}_l^T\mathbf{V}_l \quad (2.13)$$

To obtain the update rules for  $\mathbf{U}$  and  $\mathbf{V}$ , we solve the equations  $\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = 0$  and  $\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = 0$  in terms of  $\Phi$  and  $\Psi$ , respectively, and apply the Karush-Kuhn-Tucker (KKT) conditions  $\phi_{ik}u_{ik} = 0$  and  $\psi_{jk}v_{jk} = 0$  [BV09]. Since there is no Lagrange multiplier for  $\mathbf{A}$ , its value can be achieved directly by solving the equation  $\frac{\partial \mathcal{L}}{\partial \mathbf{A}} = 0$ . Finally, we end up with the following update rules:

$$u_{ik} \leftarrow u_{ik} \frac{[\mathbf{X}\mathbf{V}]_{ik}}{[\mathbf{U}\mathbf{V}^T\mathbf{V}]_{ik}} \quad (2.14)$$

$$v_{jk} \leftarrow v_{jk} \frac{[\mathbf{X}^T\mathbf{U} + \alpha(\mathbf{V}_l\mathbf{A}^T\mathbf{A})^- + \alpha(\mathbf{Q}^T\mathbf{A})^+]_{jk}}{[\mathbf{V}\mathbf{U}^T\mathbf{U} + \alpha(\mathbf{V}_l\mathbf{A}^T\mathbf{A})^+ + \alpha(\mathbf{Q}^T\mathbf{A})^-]_{jk}} \quad (2.15)$$

$$\mathbf{A} \leftarrow \mathbf{Q}\mathbf{V}_l(\mathbf{V}_l^T\mathbf{V}_l)^{-1}, \quad (2.16)$$

where for a matrix  $\mathbf{M}$ , we define  $\mathbf{M}^+$ ,  $\mathbf{M}^-$  as  $\mathbf{M}^+ = (|\mathbf{M}| + \mathbf{M})/2$  and  $\mathbf{M}^- = (|\mathbf{M}| - \mathbf{M})/2$ . As expected, the update rule for  $\mathbf{U}$  remains the same as in the original NMF algorithm [LS01], since the newly introduced term depends only on the variables  $\mathbf{V}$  and  $\mathbf{A}$ .

For the objective function of NMF, it is easy to check that if  $\mathbf{U}$  and  $\mathbf{V}$  are the solution, then,  $\mathbf{U}\mathbf{D}$ ,  $\mathbf{V}\mathbf{D}^{-1}$  will also form a solution for any positive diagonal matrix  $\mathbf{D}$ . To eliminate this uncertainty, it is required that the Euclidean length of each column vector in matrix  $\mathbf{U}$  (or  $\mathbf{V}$ ) to be 1 [XLG03]. The matrix  $\mathbf{V}$  (or  $\mathbf{U}$ ) will be



adjusted accordingly so that  $\mathbf{UV}^T$  does not change. This can be achieved by

$$u_{ik} \leftarrow \frac{u_{ik}}{\sqrt{\sum_i u_{ik}^2}} \quad (2.17)$$

$$v_{jk} \leftarrow \frac{v_{jk}}{\sqrt{\sum_j v_{jk}^2}} \quad (2.18)$$

Please see the Appendix B.1 for a detailed proof of the convergence of  $\mathbf{V}$  in the above update rule.

### 2.3.2 Computational complexity

To estimate the computational complexity of the proposed algorithm, the number of multiplication, addition/subtraction and division floating point operations is calculated. For the multiplications involving the matrices  $\mathbf{V}_l$ , we note that only the first  $N_l$  rows are relevant, since the others are 0. Therefore, the term  $\mathbf{V}_l^T \mathbf{V}_l$  needs  $N_l K^2$  multiplication and addition operations. Furthermore, for the terms involving the matrix  $\mathbf{Q}$ , we take advantage of the property that in each column of  $\mathbf{Q}$ , only one entry is different from 0. Thus, the term  $\mathbf{Q}^T \mathbf{A}$  requires  $N_l K$  multiplication operations. The computation of the term  $\mathbf{V}_l (\mathbf{V}_l^T \mathbf{V}_l)^{-1}$  can be performed efficiently with the QR-decomposition of  $\mathbf{V}_l$  [GV96] and subsequent back and forward substitution. This requires approximately  $2N_l K^2 - 2K^3/3 + K(K-1)/2$  multiplication and addition operation(s) and  $K$  division operation(s). The total number of operations for the update rules of the two algorithms are summarized in Table 2.1. It confirms that while the number of operations increases for the proposed algorithm, the overall computational complexity remains  $O(MNK)$ .

Table 2.1: Computational complexity for each iteration of NMF and DNMF

Method	multiplication	addition/subtraction	division	overall
NMF	$2MNK + 2(M+N)K^2 + (M+N)K$	$2MNK + 2(M+N)K^2$	$(M+N)K$	$O(MNK)$
DNMF	$2MNK + 2(M+N)K^2 + (M+N)K - 2/3K^3 + (3N_l + S)K^2 + 6N_l K + K(K-1)/2$	$2MNK + 2(M+N)K^2 - 2/3K^3 + (3N_l + S)K^2 + 4N_l K + K(K-1)/2$	$(M+N)K + K$	$O(MNK)$

### 2.3.3 Experiment 1

In order to understand how the proposed algorithm (DNMF) differs from other mentioned semi-supervised algorithms (i.e., CNMF and LCA), an experiment on synthetic data was performed. The data set consists of two noisy parabolas with

100 samples per parabola and each parabola corresponds to one class. We apply the algorithms to transform the dataset from the original 2D space to another 2D space, provided the label information of part of the data is available. The resulting distributions are shown in Figure 2.1, where the first and second rows present the results of CNMF, and DNMF, respectively. The first, second and third columns show the results of experiments when 0%, 40%, 100% label information is provided, respectively. Evidently, CNMF merges the labeled data points into a single point, resulting in the reduction of data points. In contrast, in DNMF, the number of samples remains the same in the new space, which is a big advantage over CNMF method.

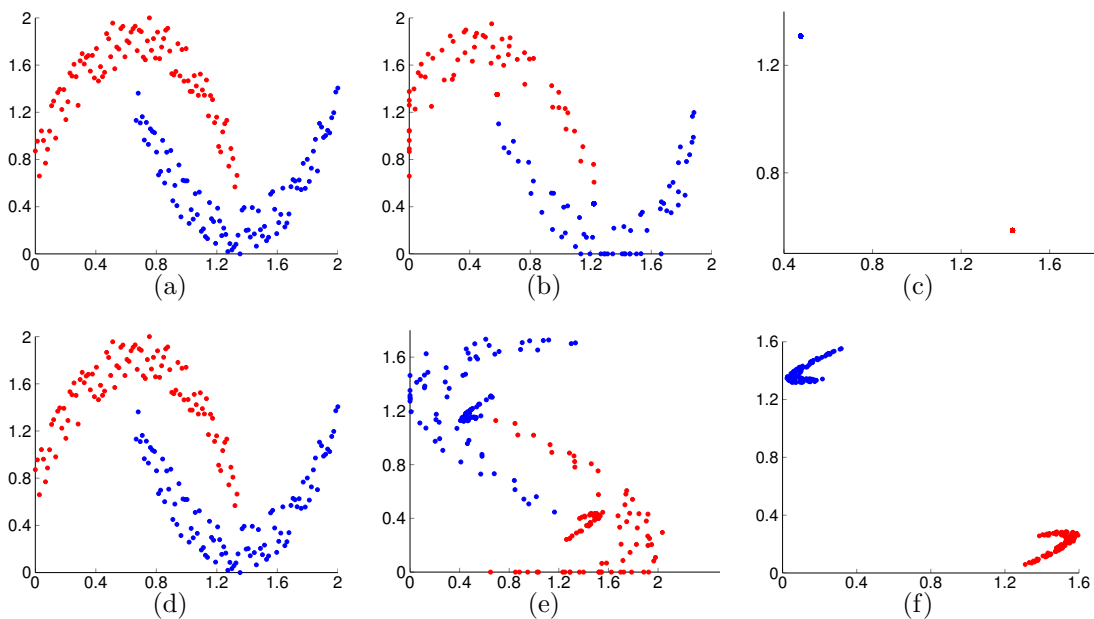


Figure 2.1: Application of CNMF, and DNMF on two-parabolas data set with different degrees of labeling; each row corresponds to one method. Each column represents the result of the same labeling degree; (a,d) original samples; (b,e) results of 40% labeling; (c,f) results of 100% labeling.

## 2.3.4 Experiment 2

### 2.3.4.1 Data sets

The experiments were performed on three data sets: 1) Yale Faces; 2) Handwritten Digits; 3) PIE Faces.

**Yale Faces** data set [Cai+06] contains  $32 \times 32$  gray scale images of the faces of 15 individuals with 11 images per person. Each image is with a different configuration or facial expression. In total, we have 165 1024-dimensional samples.

**Handwritten Digits** data set contains 10000 gray scale images of handwritten digits from 0-9 with 1000 images per class. The size of each image is  $16 \times 16$  pixels, which leads to 256-dimensional feature vectors.

**PIE Faces** data set [Cai+06] contains  $32 \times 32$  gray scale faces images of 68 people, with 42 facial images per person. Thus, in total we have 2856 1024-dimensional samples. Some example images from the Yale Faces and PIE face data sets are depicted in Figure 2.2.

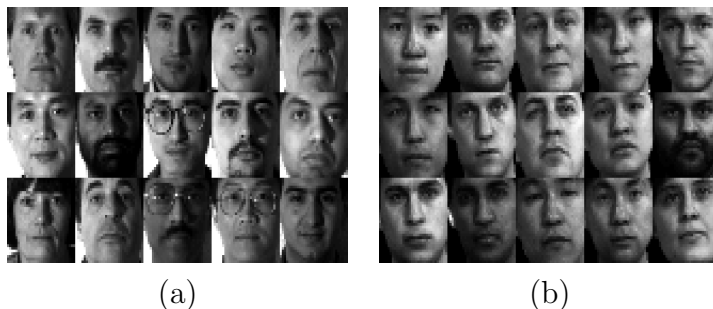


Figure 2.2: (a) The Yale Faces data set; (b) The PIE Faces data set.

#### 2.3.4.2 Evaluation metrics

To assess the quality of new features, we use them in k-means clustering to compare their results with other features. Therefore, we need evaluation metrics to assess the performance of clustering. The two metrics used in this thesis to assess the performance of clustering are (1) Accuracy (AC), and (2) normalized Mutual Information (nMI)[XLG03; Bab+14c]. The accuracy computes the percentage of correctly predicted groups, compared to the true labels and normalized mutual information measures the similarity of two clusters.

**Accuracy** represents the percentage of correctly predicted labels compared to the ground truth labels. Given a data set with  $N$  samples, where for each sample,  $t_i$  indicates its true label given by the data set and  $p_i$  is the label predicted by a clustering algorithm, the accuracy is defined as

$$AC = \frac{\sum_{i=1}^N \delta(t_i, \text{map}(p_i))}{N}, \quad (2.19)$$

where  $\delta(x, y) = 1$  if  $x = y$  and 0 otherwise, and  $\text{map}(p_i)$  is a function that maps each label to the corresponding label in the data set. The permutation mapping is determined using the Kuhn-Munkres (KM) algorithm [Kuh55].

**Normalized mutual information** determines the similarity of two clusters. Given two sets of clusters  $C = \{c_1, \dots, c_k\}$  and  $\hat{C} = \{\hat{c}_1, \dots, \hat{c}_k\}$ , the mutual information metric is computed by

$$\text{MI}(C, \hat{C}) = \sum_{c_i \in C, \hat{c}_j \in \hat{C}} p(c_i, \hat{c}_j) \log \frac{p(c_i, \hat{c}_j)}{p(c_i)p(\hat{c}_j)}, \quad (2.20)$$

where  $p(c_i), p(\hat{c}_j)$  represent the probability that an arbitrarily selected data point belongs to the clusters  $C$  or  $\hat{C}_j$ , respectively, and  $p(c_i, \hat{c}_j)$  represents the joint probability that a point belongs to both clusters simultaneously. As the similarity of the two clusters increases, the mutual information  $\text{MI}(C, \hat{C})$  increases from 0 to  $\max \{H(C), H(\hat{C})\}$ .  $H(\cdot)$  is entropy function that means  $H(C), H(\hat{C})$  represent the entropy of the clusters  $C, \hat{C}$  respectively. Dividing the mutual information by  $\max \{H(C), H(\hat{C})\}$  leads to the normalized mutual information, which takes values between 0 and 1:

$$\text{nMI}(C, \hat{C}) = \frac{\text{MI}(C, \hat{C})}{\max \{H(C), H(\hat{C})\}}. \quad (2.21)$$

### 2.3.4.3 Compared algorithms

The performance of the DNMF algorithm in generating new features is compared with several algorithms by assessing the applied k-means clustering on the computed features and measure the quality of clustering. The compared algorithms are:

- Original representation (features)
- PCA [Jol05]. PCA is a well-known dimensionality reduction algorithm. It is expected that if the reduced dimension is set to the number of classes, each class is aligned along one principal axis.
- NMF in Frobenius-Norm formulation [LS01]. Similar to PCA, if we set the reduced dimension equal to the number of classes, we expect the samples of each class to be aligned along one dimension.
- GNMF in Frobenius-Norm formulation [Cai+11]. This algorithm extends the NMF-algorithm with a similarity term, which forces samples that are close to each other in the original representation to be also close to each other in the new representation. GNMF aims to preserve the locality of the data.
- CNMF in Frobenius-Norm formulation [Hai+12]. This algorithm is a semi-supervised algorithm representing the samples with the same known class as the same point in the new representation.
- The proposed algorithm (i.e., DNMF).

#### 2.3.4.4 Clustering results

In order to compare the performance of the algorithms in clustering, the experiments were conducted with different numbers of classes,  $k$ , extracted from each data set. To obtain representative results, we repeated the experiments 10 times for each  $k$ , by selecting a random subset of  $k$  classes from the data set and computing the average results. For the dimensionality reduction techniques, we always set the new dimension equal to the number of classes and applied  $k$ -means in the new representation. Then, we evaluated the clustering results for all algorithms with the introduced metrics (i.e., AC, nMI). The  $k$ -means was repeated 20 times in each experiment and the best result was selected. To choose the proper parameters, we performed cross-validation on all algorithms and the parameter with the best results was selected for each data set.

Figure 2.3, Figure 2.4, and Figure 2.5 show the clustering results for the Yale faces, Handwritten digit, and PIE faces datasets, respectively. For each dataset, the first row shows the clustering accuracy and the second row the normalized mutual information for labeling percentages of 30%, 50% and 70%. For the case of 50% labeling, the clustering results are additionally presented in Table 2.2, Table 2.3 and Table 2.4 for the three datasets, respectively.

#### 2.3.4.5 Discussion

The results confirm that the proposed algorithm outperforms the other algorithms in most cases, especially in terms of accuracy. Moreover, by increasing label information, the difference in performance between the unsupervised and semi-supervised algorithms becomes bigger, as expected. As Figure 2.3 shows, DNMF absolutely outperforms another semi-supervised technique (i.e, CNMF). For the Handwritten digit data set, DNMF still outperforms the others in terms of accuracy, especially with high degree of labeling. However, it shows comparable results with CNMF in terms of nMI. Generally, DNMF has better performance than CNMF. For the PIE faces data set, DNMF has most of the time absolute performance in terms of accuracy. But, for the nMI, CNMF shows better performance. In Section 2.3.7, the locality preserving of DNMF for these three data sets is studied. There, we see that for both the PIE faces data set and the Handwritten digit data set, the higher degree of labeling leads to better performance in preserving the locality. However, this is not true for Yale face dataset.

#### 2.3.5 Convergence study

Here we analyze the convergence speed of the proposed algorithm and compare it with the original NMF algorithm. Figure 2.6 depicts the converge plots of NMF

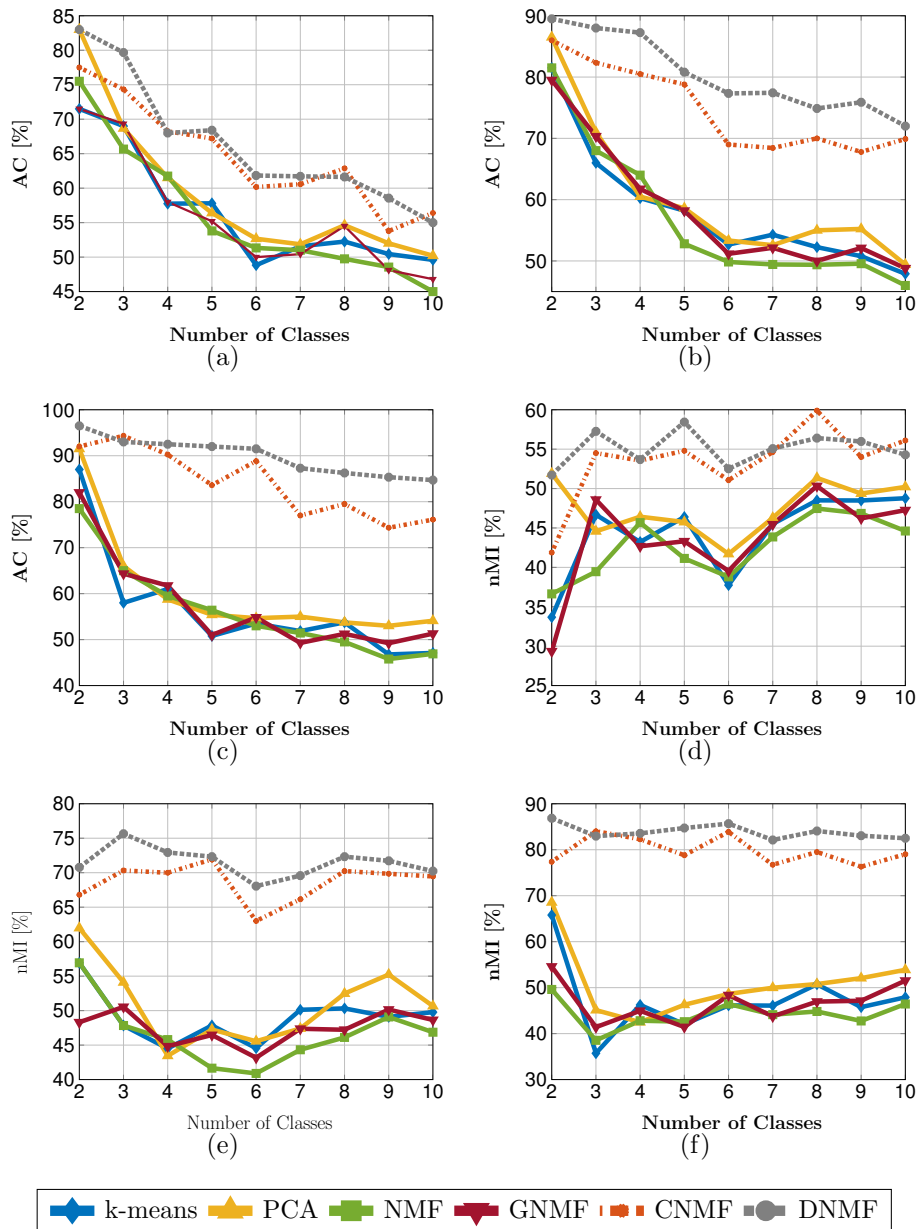


Figure 2.3: Clustering results for Yale Faces dataset. First row shows clustering accuracy for different percentages of labeling: (a) 30%; (b) 50%; (c) 70%. Second row shows normalized mutual information for different percentages of labeling: (d) 30%; (e) 50%; (f) 70%.

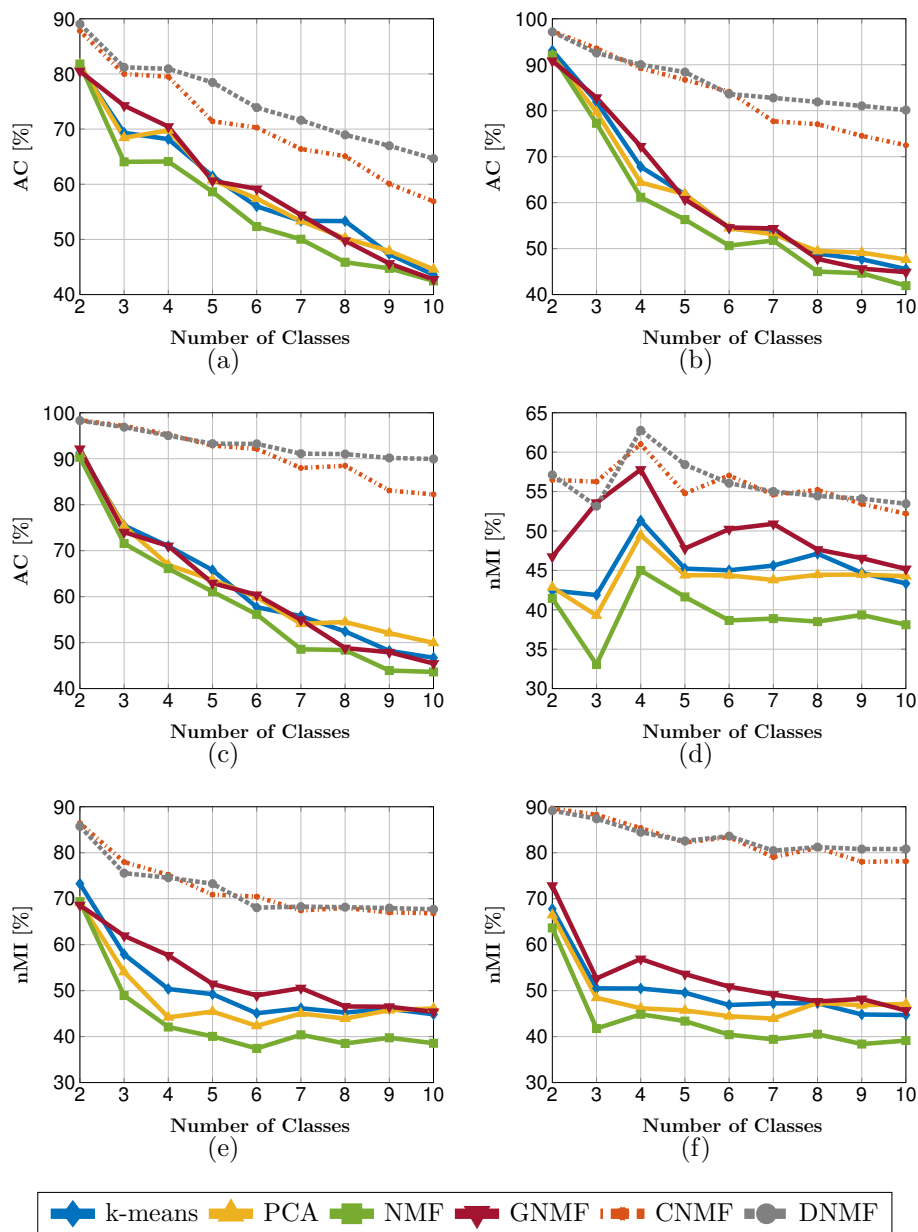


Figure 2.4: Clustering results for Handwritten Digits data set. First row shows clustering accuracy for different percentages of labeling: (a) 30%; (b) 50%; (c) 70%. Second row shows normalized mutual information for different percentages of labeling: (d) 30%; (e) 50%; (f) 70%.

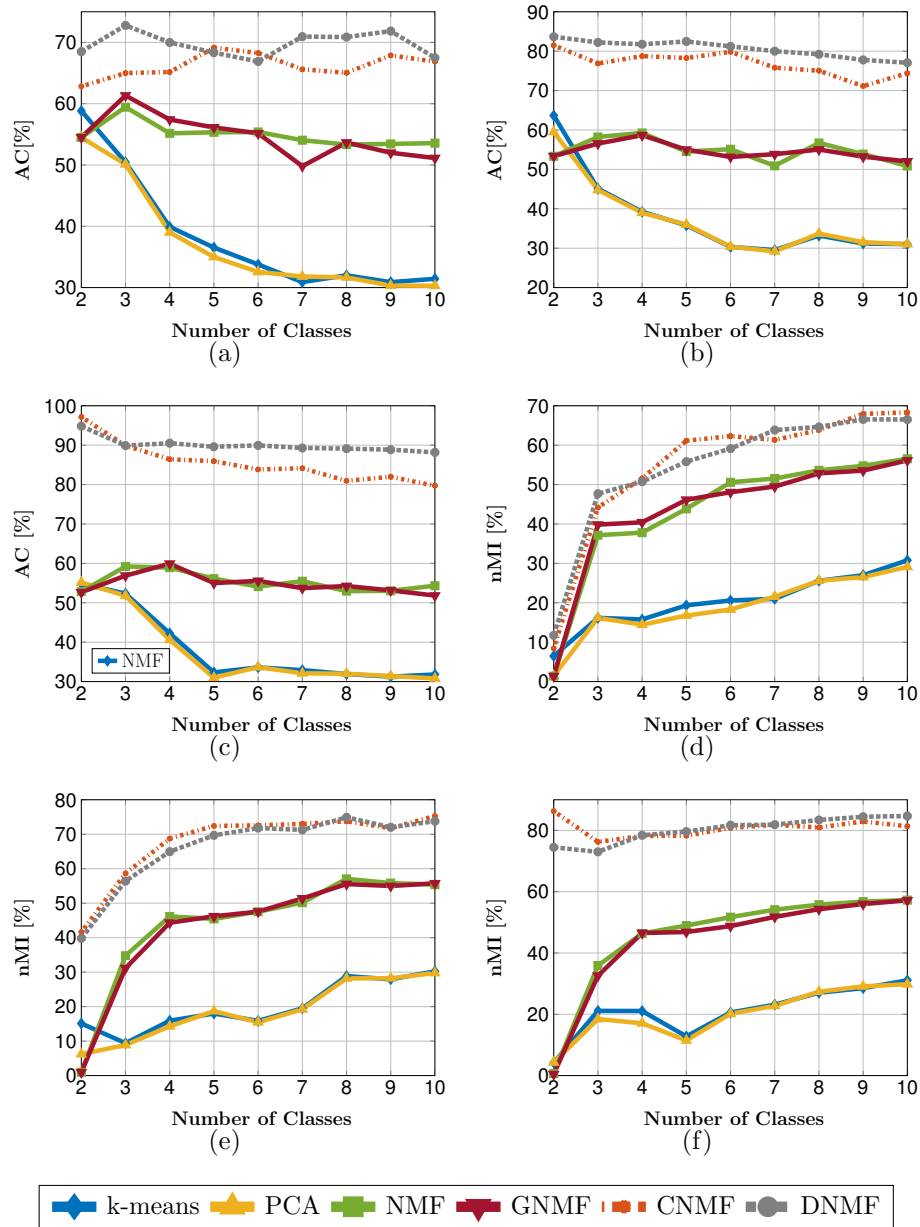


Figure 2.5: Clustering results for PIE Faces dataset. First row shows clustering accuracy for different percentages of labeling: (a) 30%; (b) 50%; (c) 70%. Second row shows normalized mutual information for different percentages of labeling: (d) 30%; (e) 50%; (f) 70%.



Table 2.2: Clustering Results for Yale Faces dataset and 50% labeling: AC (%)

Accuracy(%)						
k	k-means	PCA	NMF	GNMF	CNMF	DNMF
2	81.5 ± 22.9	86.5 ± 17.8	81.5 ± 22.9	79.5 ± 21.5	86.0 ± 20.5	<b>89.5 ± 16.2</b>
3	66.0 ± 16.7	71.0 ± 15.5	68.0 ± 10.6	70.3 ± 13.0	82.3 ± 15.0	<b>88.0 ± 12.9</b>
4	60.3 ± 15.7	60.5 ± 12.5	64.0 ± 14.0	61.8 ± 13.8	80.5 ± 11.1	<b>87.3 ± 7.0</b>
5	58.2 ± 12.8	58.6 ± 8.8	52.8 ± 9.2	58.2 ± 9.1	78.8 ± 8.4	<b>80.8 ± 8.5</b>
6	52.7 ± 9.9	53.3 ± 8.3	49.8 ± 9.0	51.2 ± 9.6	69.0 ± 11.3	<b>77.3 ± 8.2</b>
7	54.3 ± 10.0	52.6 ± 7.9	49.4 ± 7.7	52.1 ± 7.2	68.4 ± 5.8	<b>77.4 ± 5.1</b>
8	52.3 ± 6.1	55.0 ± 6.7	49.4 ± 6.1	50.0 ± 5.3	70.0 ± 5.5	<b>74.9 ± 8.1</b>
9	50.8 ± 8.6	55.2 ± 9.5	49.6 ± 7.9	52.1 ± 10.1	67.8 ± 5.6	<b>75.9 ± 7.0</b>
10	47.9 ± 7.1	49.4 ± 8.5	46.0 ± 7.0	48.8 ± 6.7	69.9 ± 6.1	<b>72.0 ± 5.5</b>
normalized Mutual Information(%)						
2	56.9 ± 47.2	62.0 ± 39.8	56.9 ± 47.2	48.3 ± 42.6	66.8 ± 41.9	<b>70.8 ± 36.4</b>
3	47.8 ± 22.8	54.1 ± 21.7	47.8 ± 18.0	50.5 ± 16.8	70.3 ± 20.0	<b>75.7 ± 20.7</b>
4	44.5 ± 19.6	43.4 ± 16.6	45.8 ± 18.1	44.8 ± 20.0	70.0 ± 14.1	<b>72.9 ± 14.2</b>
5	47.9 ± 17.4	47.3 ± 11.7	41.7 ± 10.4	46.5 ± 13.1	71.9 ± 9.3	<b>72.3 ± 7.6</b>
6	44.6 ± 11.0	45.6 ± 10.4	40.9 ± 10.0	43.2 ± 12.5	63.0 ± 10.2	<b>68.1 ± 10.7</b>
7	50.1 ± 12.7	47.4 ± 10.2	44.3 ± 8.1	47.4 ± 9.9	66.2 ± 5.8	<b>69.6 ± 7.6</b>
8	50.3 ± 7.9	52.5 ± 6.9	46.1 ± 5.1	47.2 ± 6.1	70.2 ± 5.0	<b>72.3 ± 7.3</b>
9	49.1 ± 10.1	55.2 ± 9.2	49.1 ± 8.8	50.2 ± 9.4	69.8 ± 4.3	<b>71.7 ± 6.1</b>
10	49.8 ± 8.0	50.7 ± 7.6	46.9 ± 6.3	48.6 ± 7.0	69.5 ± 4.4	<b>70.2 ± 5.0</b>

and DNMF algorithms on the three data sets. It is clear from the plots that the convergence speed of the proposed algorithm is comparable with NMF algorithm and also the algorithm converges after 100 iterations.

### 2.3.6 Parameter analysis

The performance of the proposed algorithm is controlled by the parameter  $\alpha$ . By looking at (2.8), we expect the magnitude of the optimal parameter  $\alpha$  to be dependent on the relative size of the two terms. Specifically, the size of the first term is on the order of MN and the size of the second term on the order of  $SN_l$ . In order to keep the relative weight of the two terms similar among different datasets, we therefore propose to set  $\alpha = \frac{M}{SP} \hat{\alpha}$ , where  $\mathbf{P}$  indicates the percentage of labeled samples. Then, the relative weight of the two terms is controlled by the normalized parameter  $\hat{\alpha}$ . Figure 2.7 shows the performance of the proposed algorithm on the three data sets for different values of  $\hat{\alpha}$ . Clearly, the best performance is achieved in all cases, when  $\hat{\alpha}$  is of order  $10^1$ .

Table 2.3: Clustering Results for Handwritten Digits data set and 50% labeling: AC (%)

Accuracy(%)						
k	k-means	PCA	NMF	GNMF	CNMF	DNMF
2	93.1 ± 10.0	92.0 ± 11.1	92.0 ± 11.2	90.8 ± 11.8	<b>97.3 ± 4.3</b>	97.2 ± 4.4
3	81.9 ± 9.6	79.7 ± 10.4	77.3 ± 8.5	82.8 ± 9.9	<b>93.6 ± 2.6</b>	92.6 ± 4.6
4	67.8 ± 9.2	64.4 ± 9.7	61.2 ± 11.3	72.3 ± 13.2	89.3 ± 6.8	<b>90.0 ± 3.8</b>
5	61.8 ± 4.9	61.8 ± 4.5	56.3 ± 6.8	60.7 ± 7.9	86.7 ± 1.5	<b>88.4 ± 2.1</b>
6	54.5 ± 6.7	54.5 ± 6.9	50.6 ± 6.5	54.6 ± 9.3	<b>84.1 ± 5.5</b>	83.6 ± 3.1
7	53.4 ± 4.5	53.1 ± 5.1	51.8 ± 5.8	54.4 ± 3.2	77.7 ± 8.5	<b>82.8 ± 2.8</b>
8	48.9 ± 4.9	49.5 ± 4.9	45.0 ± 2.7	47.8 ± 2.8	77.1 ± 6.3	<b>81.9 ± 1.7</b>
9	47.7 ± 3.8	49.1 ± 5.7	44.6 ± 4.6	45.7 ± 4.4	74.5 ± 4.1	<b>81.0 ± 1.3</b>
10	45.6 ± 3.0	47.6 ± 2.8	41.9 ± 2.8	44.9 ± 5.0	72.5 ± 3.6	<b>80.1 ± 1.9</b>
normalized Mutual Information(%)						
2	73.3 ± 26.5	69.2 ± 27.3	69.4 ± 28.4	68.6 ± 35.8	<b>86.4 ± 17.5</b>	85.8 ± 17.8
3	57.9 ± 11.3	54.1 ± 10.7	48.9 ± 11.5	61.9 ± 15.2	<b>78.0 ± 7.2</b>	75.6 ± 11.2
4	50.4 ± 11.8	44.2 ± 9.3	42.1 ± 9.7	57.7 ± 13.1	<b>75.2 ± 8.3</b>	74.6 ± 6.7
5	49.2 ± 6.8	45.4 ± 7.2	40.1 ± 5.6	51.5 ± 8.0	70.9 ± 2.1	<b>73.3 ± 3.1</b>
6	45.1 ± 6.7	42.3 ± 4.5	37.4 ± 4.3	48.9 ± 10.9	<b>70.5 ± 4.7</b>	68.0 ± 4.0
7	46.2 ± 3.4	45.0 ± 2.4	40.4 ± 3.5	50.5 ± 2.0	67.4 ± 5.2	<b>68.3 ± 3.5</b>
8	45.2 ± 3.2	43.9 ± 3.4	38.5 ± 2.7	46.6 ± 2.9	68.0 ± 3.8	<b>68.2 ± 2.5</b>
9	46.0 ± 3.0	45.7 ± 4.2	39.7 ± 4.0	46.5 ± 3.9	67.0 ± 3.2	<b>68.0 ± 2.2</b>
10	44.9 ± 2.4	46.1 ± 2.6	38.6 ± 2.9	45.4 ± 2.3	66.8 ± 2.6	<b>67.7 ± 2.4</b>

### 2.3.7 Locality preservation

For the locality preserving property of the DNMF algorithm, we compute the 5 nearest neighbors of each data point before and after applying the algorithm. We call the average similarity of neighborhoods over all data points the locality preserving property. DNMF, with different degrees of labeling, and GNMF are applied to all data sets and their percentages of locality preservation are depicted in Figure 2.8. Here, the experiments are repeated 10 times for different random subsets of  $k$  classes and the average results are computed. In all the data sets, DNMF is weaker than GNMF in terms of locality preservation. This makes sense, since DNMF focuses on increasing the discriminative property and GNMF focuses on locality preservation. However, for the Yale faces data set, increase of the number of classes decreases the locality preserving property, but increase of the amount of label information decreases the locality preserving property. This confirms that for this data set, the local points belong to different classes. This phenomenon is reversed for the Handwritten digit and the PIE faces data sets, where increase of the number of classes and also the degree of labeling increases the locality preservation. Thus, in this case, locality preserving has some correlation with discrimination. For the PIE faces data set, increasing the number of classes does not change the locality preservation.

Table 2.4: Clustering Results for PIE Faces dataset and 50% labeling: AC (%)

Accuracy(%)						
k	k-means	PCA	NMF	GNMF	CNMF	DNMF
2	63.7 ± 15.5	59.5 ± 10.9	53.3 ± 4.8	53.3 ± 4.8	81.5 ± 14.2	<b>83.7 ± 7.4</b>
3	45.1 ± 6.7	44.8 ± 6.6	58.2 ± 4.9	56.6 ± 4.5	76.9 ± 12.4	<b>82.2 ± 3.2</b>
4	39.3 ± 9.0	39.0 ± 8.6	59.3 ± 6.5	58.7 ± 5.9	78.7 ± 10.3	<b>81.8 ± 5.1</b>
5	35.8 ± 7.3	36.0 ± 6.8	54.5 ± 4.6	55.0 ± 4.2	78.3 ± 8.5	<b>82.5 ± 4.4</b>
6	30.3 ± 3.0	30.4 ± 3.3	55.1 ± 6.0	53.2 ± 7.3	79.8 ± 8.2	<b>81.2 ± 5.4</b>
7	29.5 ± 4.2	29.2 ± 4.2	50.9 ± 5.0	53.9 ± 5.6	75.8 ± 5.3	<b>80.0 ± 3.5</b>
8	33.2 ± 5.0	33.7 ± 5.0	56.7 ± 6.5	55.0 ± 6.4	75.0 ± 6.2	<b>79.2 ± 5.1</b>
9	31.2 ± 3.9	31.5 ± 4.1	53.9 ± 7.4	53.2 ± 7.4	71.1 ± 5.0	<b>77.8 ± 5.7</b>
10	31.0 ± 2.6	31.0 ± 2.6	50.9 ± 4.2	52.0 ± 6.0	74.4 ± 3.7	<b>77.1 ± 6.8</b>
normalized Mutual Information(%)						
2	15.1 ± 19.8	6.2 ± 9.8	1.0 ± 2.1	1.0 ± 2.1	<b>41.6 ± 30.8</b>	39.8 ± 19.4
3	9.4 ± 7.7	8.9 ± 7.8	34.8 ± 7.6	31.1 ± 8.6	<b>58.6 ± 13.7</b>	56.4 ± 6.1
4	15.9 ± 14.1	14.3 ± 13.8	46.1 ± 6.2	44.3 ± 9.0	<b>68.7 ± 9.0</b>	64.9 ± 8.2
5	18.0 ± 11.7	18.7 ± 11.0	45.4 ± 9.0	46.2 ± 7.2	<b>72.4 ± 6.7</b>	69.7 ± 6.4
6	15.9 ± 3.6	15.5 ± 5.0	47.5 ± 7.3	47.5 ± 6.8	<b>72.6 ± 6.6</b>	71.8 ± 5.5
7	19.5 ± 5.3	19.2 ± 5.2	50.1 ± 5.2	51.4 ± 4.1	<b>73.0 ± 6.1</b>	71.3 ± 5.8
8	28.9 ± 6.3	28.2 ± 6.0	57.0 ± 6.8	55.5 ± 5.1	73.7 ± 5.8	<b>74.9 ± 5.8</b>
9	27.9 ± 3.8	28.2 ± 5.2	55.8 ± 5.5	55.0 ± 6.3	71.7 ± 3.4	<b>72.0 ± 4.7</b>
10	30.2 ± 3.5	29.7 ± 2.4	55.4 ± 5.1	55.8 ± 4.4	<b>75.3 ± 3.0</b>	73.8 ± 5.1

## 2.4 Attributes constrained NMF

In this section, we explain how relative attributes, instead of binary label information, can be used in the proposed DNMF to learn discriminative subspaces from the original features [Bab+15c; Bab+14d]. First, we provide the background on relative attributes.

### 2.4.1 A Review of Relative Attributes

In addition to label information, one may describe an image using some visual attributes such as if a person is ‘smiling’, but seems to be ‘serious’, or a scene looks ‘dry’, but not ‘complex’. In contrast to binary attributes (or labels), relative attributes provides much more semantic information. For instance, we can say that the person in image A seems to be smiling more than the person in image B, or the scene in image A looks drier than the scene in image B. The concept of a relative attribute was proposed for the first time in [PG11a]. Here, the authors assume that training data presents how objects/scene categories are related according to different predefined attributes. Then, a ranking function for each attribute is learned to rank the images based on the existence of the corresponding attribute. Finally, the learned ranking functions predict the relative strength of each attribute in a test

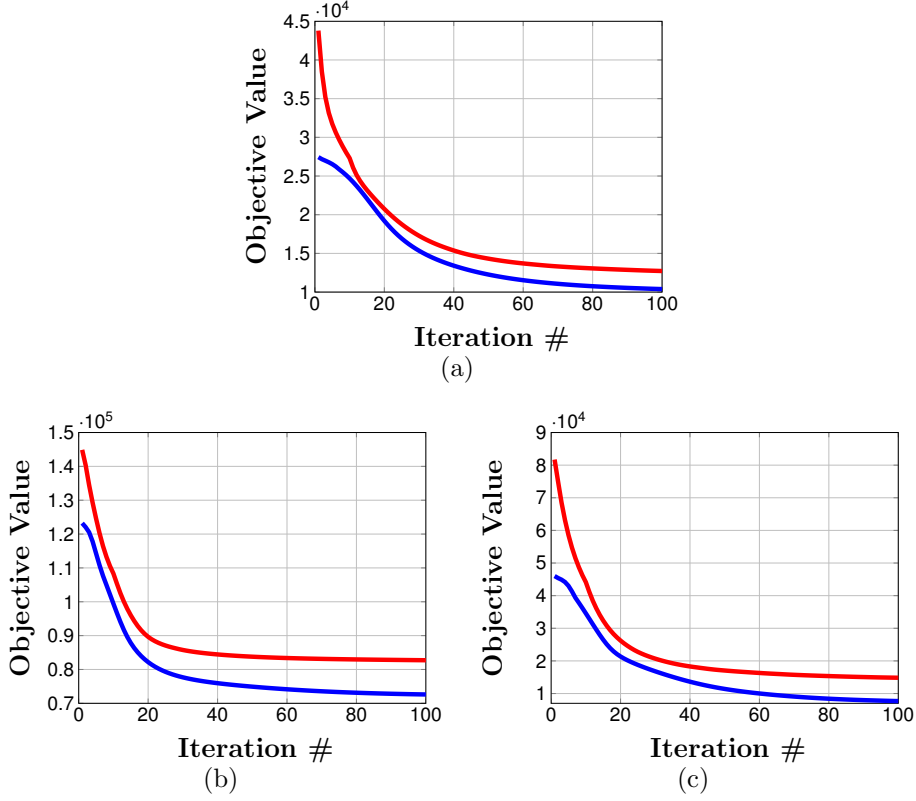


Figure 2.6: Convergence of NMF (blue) and DNMF (red) on three datasets.(a) Yale Faces; (b) Handwritten Digits; (c) PIE Faces.

image. Figure 2.9 illustrates the difference between binary and relative descriptions of images.

If there are  $M$  predefined attributes  $\mathcal{A} = \{a_m\}$ , and  $M$  ranking functions  $\mathbf{w}_m$  for  $m = 1..M$  are learned, then the predicted relative attributes are computed by

$$r_m(x_i) = \mathbf{w}_m^\top x_i, \quad (2.22)$$

such that the maximum number of the following constraints are satisfied:

$$\forall (i, j) \in \mathcal{O}_m : \mathbf{w}_m^\top \mathbf{x}_i > \mathbf{w}_m^\top \mathbf{x}_j, \quad (2.23)$$

$$\forall (i, j) \in \mathcal{S}_m : \mathbf{w}_m^\top \mathbf{x}_i \approx \mathbf{w}_m^\top \mathbf{x}_j \quad (2.24)$$

whereby  $\mathcal{O}_m = \{(i, j)\}$  includes ordered image pairs with image  $i$  containing a stronger presence of attribute  $a_m$  than image  $j$  and  $\mathcal{S}_m = \{(i, j)\}$  is a set of un-ordered pairs such that images  $i$  and  $j$  have more or less the same presence of attribute  $a_m$ . Parikh *et al.* [PG11a] proposed the following optimization problem (similar to an SVM

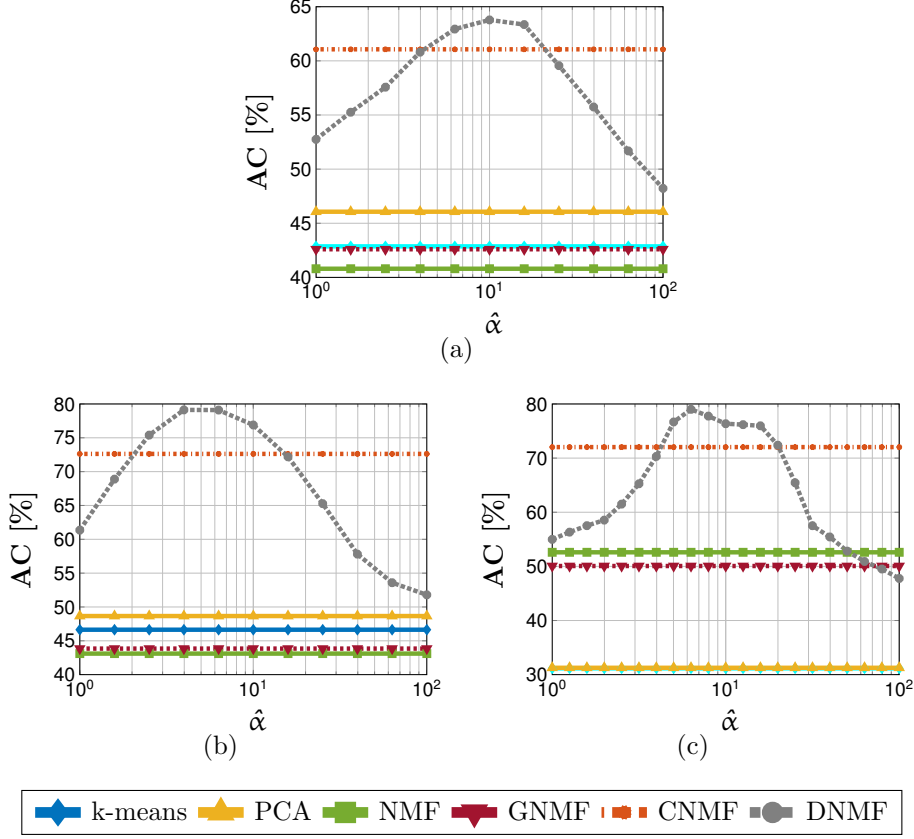


Figure 2.7: Parameter analysis of DNMF on three datasets: a) Yale faces; b) Handwritten digit; c) PIE faces.

classifier) by introducing non-negative slack variables:

$$\min \left( \frac{1}{2} \|\mathbf{w}_m^T\| + c \left( \sum \xi_{ij} + \sum \gamma_{ij} \right) \right) \quad (2.25)$$

$$\text{s.t. } \mathbf{w}_m^T(\mathbf{x}_i - \mathbf{x}_j) \geq 1 - \xi_{ij}; \quad \forall (i, j) \in \mathcal{O}_m \quad (2.26)$$

$$|\mathbf{w}_m^T(\mathbf{x}_i - \mathbf{x}_j)| \leq \gamma_{ij}; \quad \forall (i, j) \in \mathcal{S}_m \quad (2.27)$$

The solution of this optimization problem is a set of *RankSVM* functions that returns the ranking vector  $\mathbf{w}_m$  of input images and their relative order. Therefore,  $\mathbf{r}_m(\mathbf{x}_i)$  presents the relative attribute representation of image  $\mathbf{x}_i$ . By stacking the relative attributes of all input images, we build a new matrix  $\mathbf{Q}_{M \times N}$ , where M denotes the number of attributes and N is the number of images. More precisely, instead of having  $\mathbf{Q}_{S \times N}$  (S is the number of classes) we have  $\mathbf{Q}_{M \times N}$ .

In this section and the next section, we show how predicted relative attributes (instead of label information) can be used as semantic information to enhance the discriminative property of new features.

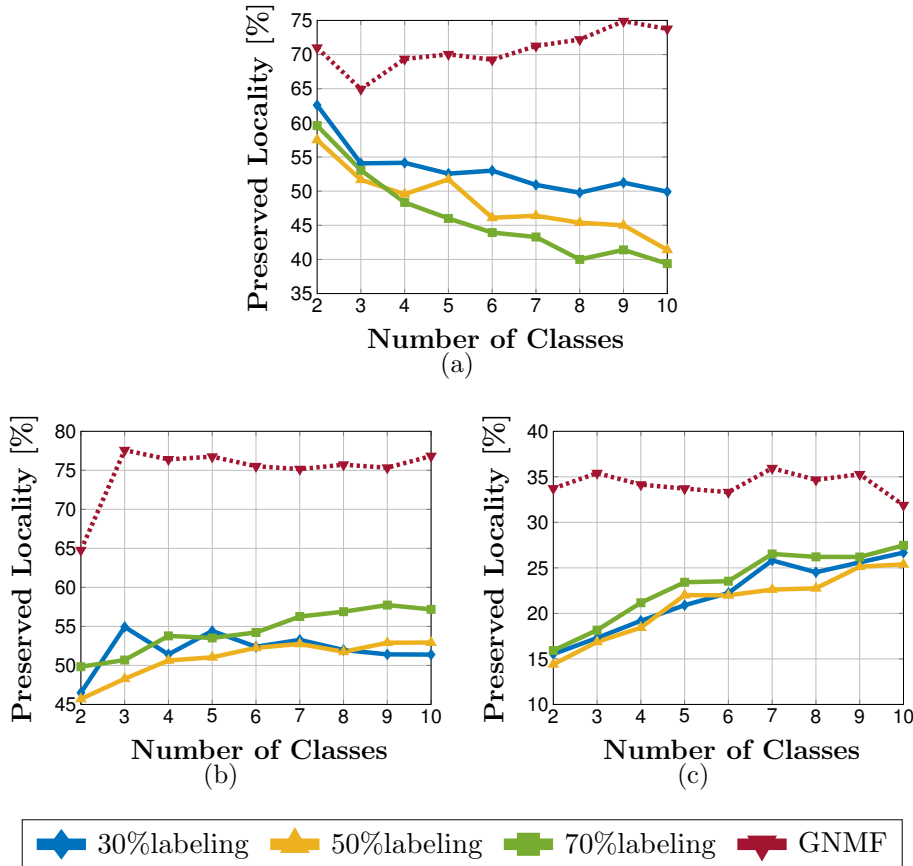


Figure 2.8: Locality preserving of DNMF with different degrees of labeling applied on the data sets: (a) Yale Faces; (b) PIE Faces; (c) Handwritten Digit.

## 2.4.2 Experiments

In this experiment, we used predicted relative attributes, instead of label information in DNMF to generate a new subspace of images. We performed our experiments by applying the proposed method to two image data sets, namely Outdoor Scene Recognition (OSR) and Public Figure Face Database (PubFig) [PG11a] (see Figure 2.10). The OSR data set contains 2688 images from 8 categories and the PubFig data set contains 772 images from 8 different individuals. The OSR images are represented by 512-dimensional GIST [OT01a] features, while PubFig images are represented by a concatenation of GIST descriptors and a 45-dimensional Lab color histogram [PG11a]. We also utilized the learned relative attributes for both data sets from [PG11a]. In 2.11, the list of predefined attributes for each data set is provided.




Image	Binary descriptions	Relative descriptions
	not natural not open perspective	more natural than tallbuilding, less natural than forest more open than tallbuilding, less open than coast more perspective than tallbuilding
	not natural not open perspective	more natural than insidicity, less natural than highway more open than street, less open than coast more perspective than highway, less perspective than insidicity
	natural open perspective	more natural than tallbuilding, less natural than mountain more open than mountain less perspective than opencountry

Figure 2.9: Binary description of images versus their relative descriptions [PG11a]

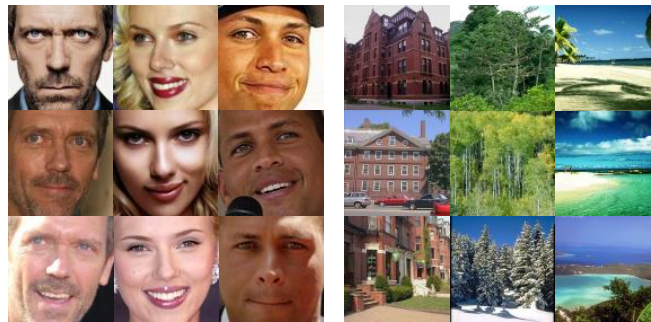


Figure 2.10: Example images from the PubFig and OSR data sets.

### 2.4.2.1 Setting

We applied the proposed method (DNMF), PCA, and NMF to the original representations of both data sets to generate (learn) different subspaces of the data. Then we applied k-means clustering, with k equal to the dimension of subspaces, to the new subspaces and also to the original features. We performed the experiments with k different classes sampled from each data set. In order to obtain representative results, we repeated the experiments 10 times for each k. The k-means runs 20 times per experiment and the best result was selected. For the subspace learning techniques (i.e., PCA, NMF, DNMF), we always set the dimension of the subspace equal to the number of classes. In DNMF, the regularization parameter was chosen by running a cross-validation on each data set. For OSR and PubFig, this parameter was 10 and 100, respectively.

	Binary	Relative
OSR	TI SHC OMF	
natural	0 0 0 0 1 1 1 1	T<I~S<H<C~O~M~F
open	0 0 0 1 1 1 1 0	T~F<I~S<M<H~C~O
perspective	1 1 1 1 0 0 0 0	O<C<M~F<H<I<S<T
large-objects	1 1 1 0 0 0 0 0	F<O~M<I~S<H~C<T
diagonal-plane	1 1 1 1 0 0 0 0	F<O~M<C<I~S<H<T
close-depth	1 1 1 1 0 0 0 1	C<M<O<T~I~S~H~F
PubFig	ACHJ MS VZ	
Masculine-looking	1 1 1 1 0 0 1 1	S<M<Z<V<J<A<H<C
White	0 1 1 1 1 1 1 1	A<C<H<Z<J<S<M<V
Young	0 0 0 0 1 1 0 1	V<H<C<J<A<S<Z<M
Smiling	1 1 1 0 1 1 0 1	J<V<H<A~C<S~Z<M
Chubby	1 0 0 0 0 0 0 0	V<J<H<C<Z<M<S<A
Visible-forehead	1 1 1 0 1 1 1 0	J<Z<M<S<A~C~H~V
Bushy-eyebrows	0 1 0 1 0 0 0 0	M<S<Z<V<H<A<C<J
Narrow-eyes	0 1 1 0 0 0 1 1	M<J<S<A<H<C<V<Z
Pointy-nose	0 0 1 0 0 0 0 1	A<C<J~M~V<S<Z<H
Big-lips	1 0 0 0 1 1 0 0	H<J<V<Z<C<M<A<S
Round-face	1 0 0 0 1 1 0 0	H<V<J<C<Z<A<S<M

Figure 2.11: The list of attributes used for each data set, along with the binary and relative attribute annotations [PG11a]

### 2.4.2.2 Results

By setting the dimension of the subspace to 2, we can visualize the data sets in 2D. For OSR and PubFig data sets, the results are depicted in Figure 2.12(a) and Figure 2.12(b), respectively. Here, it is clearly observable that all images with similar attributes are located close to each other. For instance, all OSR images with the openness attribute are placed in the bottom left part of the layout.

The results of clustering applied to the learned subspaces are depicted in Figure 2.13. Figure 2.13(a) and Figure 2.13(c) show the accuracy and normalized Mutual Information (nMI) of the clustering results for the PubFig data set. The OSR results are depicted in Figure 2.13(b) and Figure 2.13(d). It can be seen that the proposed method outperforms the other techniques significantly in both data sets. For the PubFig data set, we even achieve 75% – 85% accuracy. The algorithm converges quickly after 20 iterations and therefore can be considered computationally efficient.

The experimental results confirm that the proposed method generates the subspaces with different semantic attributes successfully.

## 2.5 Attributes constrained dictionary learning

Since last decade, sparse coding has been widely used in a variety of problems in computer vision and image analysis including image denoising, image classification,





Figure 2.12: 2D visualization of the data sets computed by the proposed method (DNMF); (a) the OSR data set; (b) the PubFig data set. Images with the same attribute are located close to each other.

Table 2.5: Clustering results of different methods on the PubFig dataset

Accuracy(%)				
k	O. Feat	PCA	NMF	DNMF
2	<b>86.0 ± 14.5</b>	85.7 ± 14.6	86.0 ± 14.2	85.1 ± 16.1
3	73.4 ± 12.6	73.3 ± 12.4	72.8 ± 13.2	<b>75.4 ± 11.7</b>
4	66.2 ± 8.6	65.4 ± 7.1	71.8 ± 9.0	<b>76.7 ± 9.4</b>
5	61.9 ± 8.6	59.5 ± 10.2	62.4 ± 8.9	<b>68.2 ± 8.8</b>
6	58.1 ± 9.6	57.0 ± 9.4	61.4 ± 8.2	<b>67.5 ± 7.4</b>
7	54.9 ± 5.0	53.7 ± 4.5	58.9 ± 3.5	<b>62.1 ± 5.1</b>
8	53.5 ± 2.2	53.3 ± 2.4	58.6 ± 2.9	<b>62.1 ± 3.1</b>
normalized Mutual Information(%)				
2	52.2 ± 27.8	51.4 ± 28.1	51.5 ± 27.7	<b>53.2 ± 34.0</b>
3	43.8 ± 16.1	42.8 ± 16.0	40.7 ± 18.6	<b>48.4 ± 15.3</b>
4	41.8 ± 11.3	39.9 ± 9.4	47.0 ± 11.1	<b>55.7 ± 10.5</b>
5	42.3 ± 7.7	41.0 ± 7.2	43.1 ± 8.1	<b>50.0 ± 9.3</b>
6	41.3 ± 9.5	40.9 ± 9.4	43.4 ± 7.5	<b>49.6 ± 6.6</b>
7	41.5 ± 4.3	40.2 ± 4.1	43.4 ± 2.8	<b>46.6 ± 4.2</b>
8	42.6 ± 1.6	41.9 ± 1.1	44.9 ± 1.6	<b>48.6 ± 3.3</b>

and image restoration. K-SVD algorithm [AEB06] and the Method of Optimal Direction (MOD) [EAH99] are the first approaches proposed for Dictionary Learning (DL), where no semantic information is used in the learning process. One sub-field of dictionary learning is discriminative DL, where either the discriminative property of the signal reconstruction residual, or of the sparse representation itself is enhanced. The work of Ramirez *et al.* [RSS10], which includes a structured incoherence term

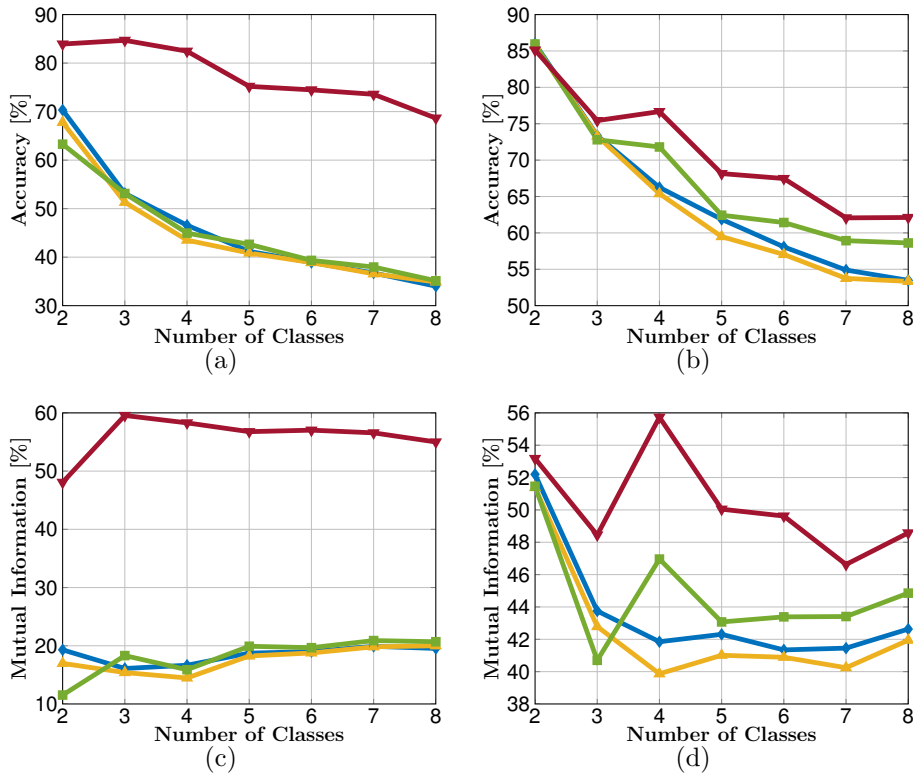


Figure 2.13: Clustering results computed by PCA (cyan), NMF (black), DNMF (blue) and original data (red) evaluated by accuracy (AC) and normalized mutual information (nMI). (a) and (b) show the AC and nMI for the OSR dataset, respectively. (c) and (d) show the AC and nMI for the PubFig dataset, respectively.

to find independent sub-directories for each class, focuses on the reconstruction residual. In the work of Gao *et al.* [GTM14] sub-dictionaries for the different classes are learned as well as a shared dictionary over all classes.

Methods aiming at finding discriminative coding vectors learn simultaneously a dictionary and a classifier. In the work of Zhang *et al.* [ZL10], the K-SVD algorithm is extended by a linear classifier. Jiang *et al.* [JLD11] propose the so called label consistent KSVD (LC-KSVD) algorithm by introducing an additional discriminative regularizer. Both of these algorithms show good results for image classification and face recognition tasks. The approach of Yang *et al.* [YZF11] combines the two types of DDL by taking into account the discriminative capabilities of the reconstruction residual and the sparse representation. Therefore, class specific sub-dictionaries are learned while maintaining discriminative coding vectors by applying the Fisher discrimination criterion. In the recent work of Cai *et al.* [Cai+14], a new Support Vector Guided Dictionary Learning (SVGDL) algorithm is presented where the discrimination term consists of a weighted summation over squared distances

Table 2.6: Clustering results of different methods on the OSR dataset

Accuracy(%)				
k	O. Feat	PCA	NMF	DNMF
2	70.3 ± 13.6	67.8 ± 14.2	63.3 ± 13.7	<b>83.9 ± 14.8</b>
3	53.1 ± 11.1	51.3 ± 9.1	53.1 ± 6.7	<b>84.7 ± 6.9</b>
4	46.6 ± 7.5	43.5 ± 6.3	45.0 ± 7.2	<b>82.4 ± 4.1</b>
5	41.2 ± 5.0	40.8 ± 4.8	42.6 ± 6.4	<b>75.2 ± 9.2</b>
6	38.9 ± 4.6	38.9 ± 4.2	39.3 ± 4.3	<b>74.5 ± 5.1</b>
7	36.7 ± 3.9	36.6 ± 3.6	38.0 ± 3.7	<b>73.6 ± 4.3</b>
8	34.0 ± 1.6	34.9 ± 1.4	35.1 ± 1.6	<b>68.7 ± 4.2</b>
normalized Mutual Information(%)				
2	19.3 ± 21.7	17.0 ± 21.7	11.5 ± 21.1	<b>48.1 ± 26.9</b>
3	16.1 ± 14.3	15.4 ± 12.3	18.3 ± 8.7	<b>59.6 ± 9.7</b>
4	16.7 ± 8.7	14.5 ± 6.6	15.9 ± 8.9	<b>58.3 ± 6.8</b>
5	18.7 ± 6.2	18.2 ± 5.8	19.9 ± 6.4	<b>56.8 ± 7.9</b>
6	19.0 ± 4.0	18.8 ± 3.7	19.7 ± 3.5	<b>57.0 ± 4.8</b>
7	19.9 ± 4.4	19.8 ± 4.0	20.9 ± 3.3	<b>56.6 ± 3.1</b>
8	19.5 ± 1.1	20.0 ± 1.7	20.7 ± 1.2	<b>55.0 ± 3.2</b>

between the pairs of coding vectors. The algorithm automatically assigns non-zero weights to critical vector pairs (the support vectors) leading to an average good performance in pattern recognition tasks.

### 2.5.1 Proposed method

We assume  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]$  to be the set of  $p$ -dimensional  $n$  input signals,  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  to be their corresponding  $k$ -dimensional sparse representation and  $\mathbf{D} \in \mathbb{R}^{n \times k}$  to be the dictionary. As a consequence, the standard dictionary learning method is defined by

$$\langle \mathbf{D}, \mathbf{X} \rangle = \underset{\mathbf{D}, \mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{DX}\|_2^2 + \lambda_1 \|\mathbf{X}\|_1, \quad (2.28)$$

with the regularization parameter  $\lambda_1$ . In order to take the relative attributes into account, the objective function has to be extended with an additional term  $\mathcal{L}(\mathbf{X})$ .

$$\langle \mathbf{D}, \mathbf{X} \rangle = \underset{\mathbf{D}, \mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{DX}\|_2^2 + \lambda_1 \|\mathbf{X}\|_1 + \lambda_2 \mathcal{L}(\mathbf{X}) \quad (2.29)$$

The *RankSVM* function maps the original input signal ( $\mathbf{y}_i$ ) to a point ( $q_i$ ) in a relative attribute space. Additionally, we assume that there exists a linear transformation (i.e.,  $\mathbf{A}$ ) that maps the sparse signal ( $\mathbf{x}_i$ ) to the point  $q_i$  (see Figure 2.14 and Eq. (2.30)). First, we define the matrix  $\mathbf{Q} \in \mathbb{R}^{N \times M}$  with the elements  $q_{im} =$

## 2. Discriminative Data Representation

$\mathbf{r}_m(\mathbf{y}_i)$ , which contains the strength of the (relative) attributes of all signals in  $\mathbf{Y}$ . In order to find the transformation of  $\mathbf{Y}$  into  $\mathbf{Q}$ , we apply the *RankSVM* function known from [PG11a] to the original input signal and obtain the weighting matrix  $\mathbf{W} = [\mathbf{w}_1^T; \mathbf{w}_2^T; \dots; \mathbf{w}_M^T]$ .

$$\operatorname{argmin}_{\mathbf{A}} \|\mathbf{Q} - \mathbf{A}\mathbf{X}\|_2^2 = \operatorname{argmin}_{\mathbf{A}} \|\mathbf{W}\mathbf{Y} - \mathbf{A}\mathbf{X}\|_2^2. \quad (2.30)$$

The objective is finding a matrix  $\mathbf{A}$ , which transforms the sparse representation

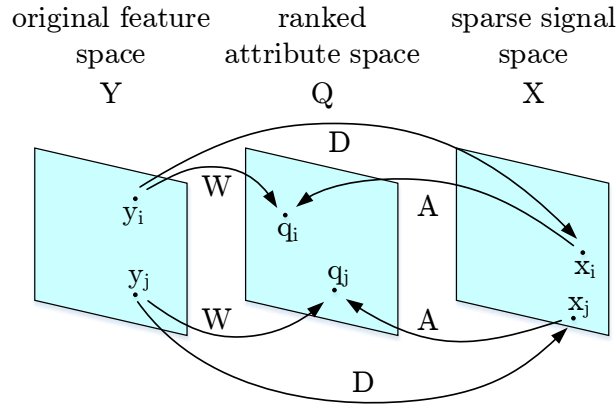


Figure 2.14: Signal transformations of  $\mathbf{x}_i$  and  $\mathbf{x}_j$  as close as possible to  $q_i$  and  $q_j$ .

of the signals into their corresponding relative attribute representations  $\mathbf{Q}$  with a minimum distance between  $\mathbf{w}_m^T \mathbf{y}_i$  and  $\mathbf{a}_m^T \mathbf{x}_i$ . By using Eq. (2.30) in Eq. (2.29) as a loss term, we get the formulation

$$\langle \mathbf{D}, \mathbf{X} \rangle = \operatorname{arg\,min}_{\mathbf{D}, \mathbf{X}, \mathbf{A}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_2^2 + \lambda_1 \|\mathbf{X}\|_1 + \lambda_2 \|\mathbf{W}\mathbf{Y} - \mathbf{A}\mathbf{X}\|_2^2. \quad (2.31)$$

From the first part of the equation, we can see that  $\mathbf{Y} \cong \mathbf{D}\mathbf{X}$ . If the  $\mathbf{Y}$  in the loss term for the relative attributes is approximated by  $\mathbf{D}\mathbf{X}$ , then the equation becomes

$$\langle \mathbf{D}, \mathbf{X} \rangle = \operatorname{arg\,min}_{\mathbf{D}, \mathbf{X}, \mathbf{A}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_2^2 + \lambda_1 \|\mathbf{X}\|_1 + \lambda_2 \|\mathbf{W}\mathbf{D}\mathbf{X} - \mathbf{A}\mathbf{X}\|_2^2. \quad (2.32)$$

The third term of Eq. (2.32) is minimized if  $\mathbf{A} = \mathbf{W}\mathbf{D}$ . This information can be used to eliminate  $\mathbf{A}$  from Eq. (2.31) to arrive at the final objective function:

$$\langle \mathbf{D}, \mathbf{X} \rangle = \operatorname{arg\,min}_{\mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_2^2 + \lambda_1 \|\mathbf{X}\|_1 + \lambda_2 \|\mathbf{W}(\mathbf{Y} - \mathbf{D}\mathbf{X})\|_2^2. \quad (2.33)$$

Additionally, we can replace the term  $\|\mathbf{X}\|_1$  with  $\|\mathbf{X}\|_2^2$ , since the goal is to learn a discriminative dictionary and not to obtain sparse signals (as in [Cai+14]). However,

once the dictionary is learned, the sparse representation is obtained by the orthogonal matching pursuit [RZE08]. Finally, we end up with the following optimization problem:

$$\langle \mathbf{D}, \mathbf{X} \rangle = \operatorname{argmin}_{\mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_2^2 + \lambda_1 \|\mathbf{X}\|_2^2 + \lambda_2 \|\mathbf{W}(\mathbf{Y} - \mathbf{DX})\|_2^2. \quad (2.34)$$

Since this equation is not a joint convex optimization problem,  $\mathbf{X}$  and  $\mathbf{D}$  are optimized sequentially. The update rules for  $\mathbf{D}$  and  $\mathbf{X}$  are found by deriving the objective function and setting the derivatives to zero.

$$O = \|\mathbf{Y} - \mathbf{DX}\|_2^2 + \lambda_1 \|\mathbf{X}\|_2^2 + \lambda_2 \|\mathbf{W}(\mathbf{Y} - \mathbf{DX})\|_2^2 \quad (2.35)$$

$$\begin{aligned} \frac{\partial O}{\partial \mathbf{D}} &= -2(\mathbf{Y} - \mathbf{DX})\mathbf{X}^T + 2\lambda_2 \mathbf{W}^T(\mathbf{WY} - \mathbf{WDX})\mathbf{X}^T = 0 \\ &= (\mathbf{Y} - \mathbf{DX}) + \lambda_2 \mathbf{W}^T \mathbf{W}(\mathbf{Y} - \mathbf{DX}) = 0 \\ &= (\mathbf{I} + \lambda_2 \mathbf{W}^T \mathbf{W})(\mathbf{Y} - \mathbf{DX}) = 0 \\ &\Rightarrow \mathbf{D} = \mathbf{Y}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \end{aligned} \quad (2.36)$$

$$\begin{aligned} \frac{\partial O}{\partial \mathbf{X}} &= -2\mathbf{D}^T(\mathbf{Y} - \mathbf{DX}) + 2\lambda_1 \mathbf{X} - 2\lambda_2 \mathbf{D}^T \mathbf{W}^T(\mathbf{WY} - \mathbf{WDX}) = 0 \\ &= (\mathbf{D}^T \mathbf{D} + \lambda_1 \mathbf{I} + \lambda_2 \mathbf{D}^T \mathbf{W}^T \mathbf{W} \mathbf{D})\mathbf{X} - \mathbf{D}^T \mathbf{Y} - \lambda_2 \mathbf{D}^T \mathbf{W}^T \mathbf{Y} = 0. \end{aligned} \quad (2.37)$$

Therefore, we have:

$$\mathbf{X} = (\mathbf{D}^T \mathbf{D} + \lambda_1 \mathbf{I} + \lambda_2 \mathbf{D}^T \mathbf{W}^T \mathbf{W} \mathbf{D})^{-1} (\mathbf{D}^T \mathbf{Y} + \lambda_2 \mathbf{D}^T \mathbf{W}^T \mathbf{Y}). \quad (2.38)$$

The algorithm works as follows. Initially the *RankSVM* [PG11a] function is used to learn the ranking matrix  $\mathbf{W}$  from the original input data  $\mathbf{Y}$  and its relative ordering (i.e., sets  $\mathcal{O}_m, \mathcal{S}_m$ ). The initial dictionary  $\mathbf{D}$  and the sparse representation of the data are obtained by first applying the K-SVD algorithm [AEB06]. Afterward, the dictionary and the sparse representation are alternately optimized until convergence. In order to avoid scaling issues, the dictionary is  $L_2$  normalized column-wise. The whole algorithm can be seen in Algorithm 1. In order to get the sparse representation from the learned dictionary, we solve the error-constrained sparse coding problem in Eq. (2.39), with the help of the OMP-Box Matlab toolbox [RZE08], where the reconstruction error from the training phase is chosen as  $\varepsilon$ . The obtained sparse signals can then be used for clustering.

$$\hat{\mathbf{X}} = \operatorname{argmin}_{\mathbf{X}} \|\mathbf{X}\|_0 \quad \text{s.t.} \quad \mathbf{X} = \|\mathbf{Y} - \mathbf{DX}\|_2^2 \leq \varepsilon, \quad (2.39)$$

**Algorithm 1** Relative Attribute Guided Dictionary Learning**Require:** Original signal  $\mathbf{Y}$ , sets of ordered ( $\mathcal{O}_m$ ) and un-ordered images ( $\mathcal{S}_m$ )**Ensure:** Dictionary  $\mathbf{D}$ 

- 1:  $\mathbf{W} \leftarrow \text{RankSVM}(\mathbf{Y}, \mathcal{O}_m, \mathcal{S}_m)$
- 2:  $\mathbf{D}_{init} \leftarrow \text{rndperm}(\mathbf{Y})$
- 3:  $\mathbf{D}, \mathbf{X} \leftarrow \text{KSVD}(\mathbf{D}_{init}, \mathbf{Y})$
- 4: **for**  $i = 0$  to numIter **do**
- 5:    $\mathbf{D} \leftarrow \mathbf{Y}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$
- 6:    $\mathbf{D} \leftarrow \text{normcol}(\mathbf{D})$
- 7:    $\mathbf{X} \leftarrow (\mathbf{D}^T \mathbf{D} + \lambda_1 \mathbf{I} + \lambda_2 \mathbf{D}^T \mathbf{W}^T \mathbf{W} \mathbf{D})^{-1} (\mathbf{D}^T \mathbf{Y} - \lambda_2 \mathbf{D}^T \mathbf{W}^T \mathbf{Y})$
- 8: **end for**

## 2.5.2 Experiments

In order to assess the quality of the learned dictionary, we propose a clustering task for the two data sets, namely PubFig [Kum+09] and OSR [OT01a]. We did cross validation and found parameters  $\lambda_1 = 0.01$  and  $\lambda_2 = 1$  optimal for all experiments and data sets. The convergence of algorithm applied to the two data sets is shown in Figure 2.15.

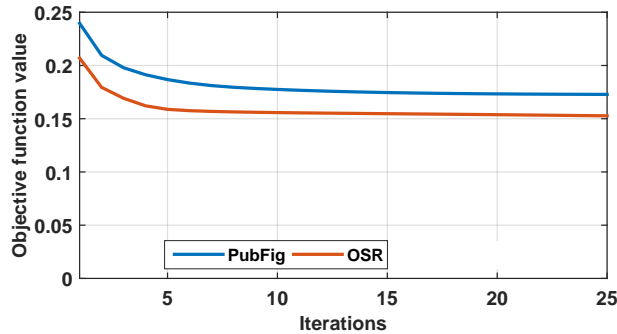


Figure 2.15: Convergence of the objective function for the two data sets

As before, we quantify the clustering performance by accuracy and normalized mutual information metrics.

### 2.5.2.1 Results

The first experiment is a comparison of Eq. (2.28) and Eq. (2.29). The proposed algorithm is applied to the two introduced data sets, once with the usage of Eq. (2.28) and once with Eq. (2.29). This experiment shows the benefit of using the additional loss term that takes the relative attributes into account. Figure 2.17 shows the

accuracy (AC) and normalized mutual information (nMI) for 100 iterations of the algorithm. A clear improvement of clustering for the PubFig and OSR data sets can be seen.

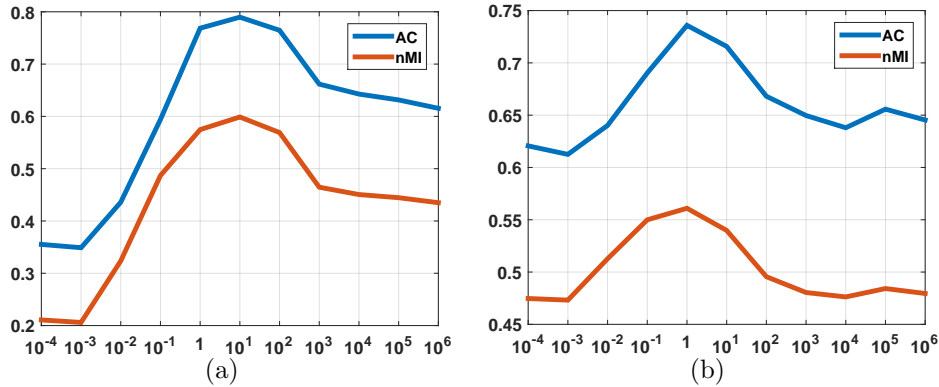


Figure 2.16: Evaluation of  $\lambda_2$  for the two data sets (from left to right, Pubfig, OSR).

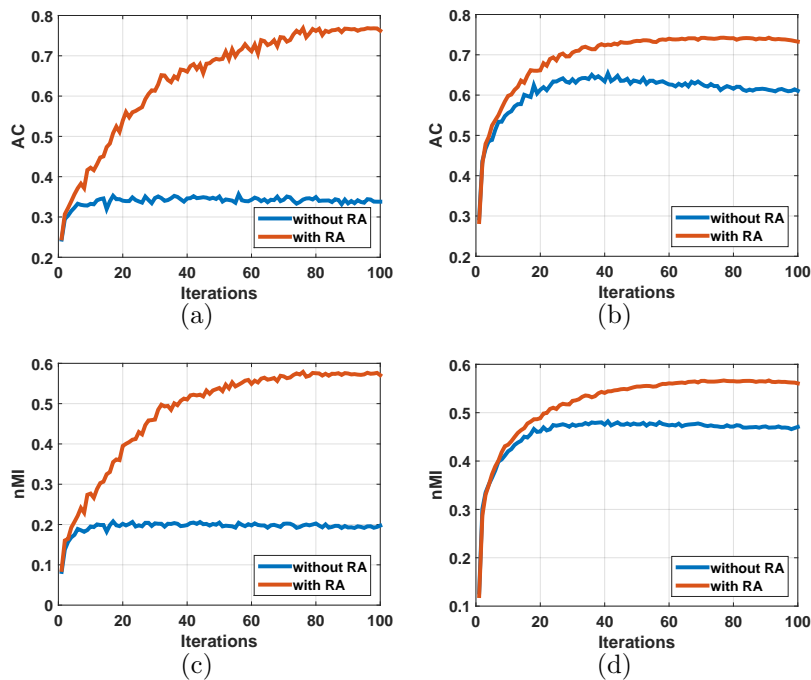


Figure 2.17: The clustering results obtained from the proposed method with and without relative attributes for the PubFig (first column) and OSR (second column) data sets. The first and second rows show the accuracy and the normalized mutual information (nMI), respectively.

As a benchmark for the results, different unsupervised and supervised (discrim-

## 2. Discriminative Data Representation

---

inative) dictionary learning techniques are used, namely (1) K-SVD [AEB06], (2) SRC [Wri+09] as unsupervised techniques, and (3) LC-KSVD [ZL10], (4) FDDL [YZF11], (5) SVGDL [Cai+14] as supervised techniques. Additionally, the original features (O. Feat.) are clustered as well by the k-means algorithm to evaluate the additional value of using relative attributes as semantic information. The results were compared on the basis of their performance for full label information, varying dictionary sizes, and a varying amount of training data. Table 2.7 shows the average accuracy and normalized mutual information for all algorithms tested on the two data sets, when using all training data, their label information, and a fixed dictionary size of 130. Evidently, although the proposed algorithm uses a different kind of semantic information, it reaches a higher performance for both data sets in comparison to other supervised and unsupervised algorithms.

Accuracy							
Method	O.Feat.	SRC	KSVD	LC-KSVD	FDDL	SVGDL	proposed
PubFig	0.324	0.226	0.310	0.306	0.584	0.595	<b>0.789</b>
OSR	0.563	0.239	0.466	0.500	0.680	0.662	<b>0.731</b>
Avg.	0.414	0.221	0.374	0.369	0.576	0.579	<b>0.661</b>
normalized Mutual Information							
PubFig	0.170	0.062	0.159	0.161	0.417	0.448	<b>0.600</b>
OSR	0.433	0.071	0.334	0.342	0.498	0.521	<b>0.564</b>
Avg.	0.308	0.065	0.251	0.241	0.441	0.459	<b>0.519</b>

Table 2.7: Accuracy and normalized mutual information for several dictionary learning algorithms applied to the data sets

Runtime (in seconds)							
Method	O.Feat.	SRC	KSVD	LC-KSVD	FDDL	SVGDL	proposed
PubFig	-	-	3.910	5.652	33.170	8.130	<b>1.443</b>
OSR	-	-	3.803	5.467	32.492	7.628	<b>1.422</b>
Avg.	-	-	4.276	6.059	31.993	8.457	<b>1.749</b>

Table 2.8: Runtime (in seconds) for several dictionary learning algorithms applied to the data sets.

In Table 2.8, the runtime of the training phase of the algorithms is analyzed, where the numbers confirm that the proposed algorithm runs much faster than all other contestants. The experiments were conducted on an Asus N56VZ-S4044V Notebook with an Intel Core i7-3610QM processor and a clock speed of 2.3 GHz and the codes for the compared algorithms were extracted from the corresponding



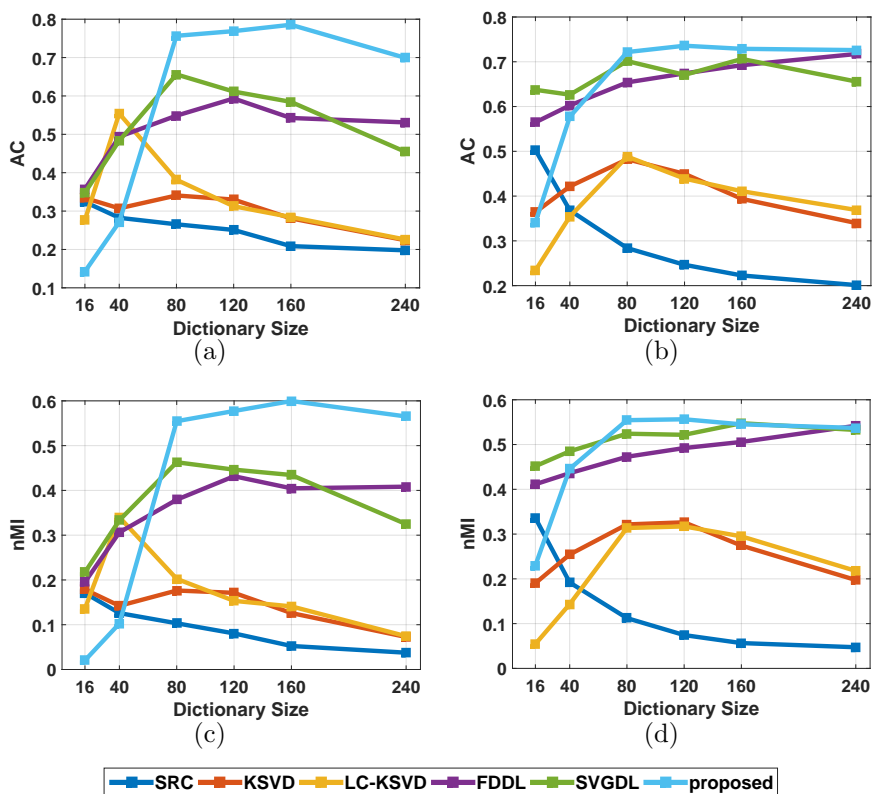


Figure 2.18: Clustering results for PubFig (first column) and OSR (second column) data sets for increasing dictionary sizes.

projects or publications pages.

Figure 2.18 shows the behavior of the algorithms for an increasing dictionary size with all available training data. The dictionary sizes used were [16, 40, 80, 120, 160, 240] for the PubFig and OSR data sets, which corresponds to [2, 5, 10, 15, 20, 30] atoms per class. The number of atoms per class are constrained by the partition of the data into training and testing. Note that the FDDL algorithm cannot use all training data, since the dictionary size restricts the size of the training samples. Therefore, only in the last test case does the algorithm use the complete training information. The results show that for the proposed algorithm the accuracy increases with the dictionary size, up to values exceeding the compared algorithms. Figure 2.19 illustrates the results when the amount of used training data is varied. In addition, the dictionary sizes were matched to the size of the training data. Again, the proposed algorithm can exceed the results of the compared approaches up to a number of training samples in the OSR data set, where the SVGDL and FDDL algorithms produce comparable results. The number of training samples per class

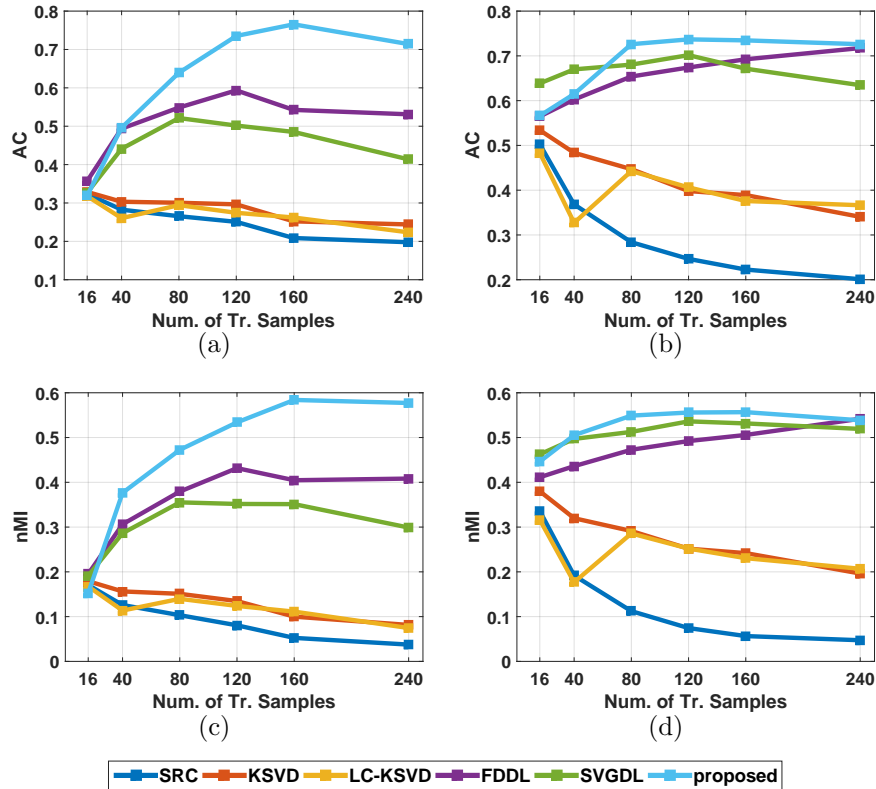


Figure 2.19: Clustering results for PubFig (first column) and OSR (second column) data sets for increasing training data.

were [2, 5, 10, 15, 20, 30] for the PubFig and OSR data sets.

## 2.6 Summary and conclusion

We presented a novel semi-supervised NMF-formulation, called DNMF. In DNMF, the new representation is formed by adding an additional constraint that enforces samples with the same class to be aligned on the same axis in the new representation. In contrast to other semi-supervised methods, this approach does not merge data points with the same label into a single point. We showed the DNMF approach in the F-norm formulation and proposed update rules to solve the optimization problems. Experimental results on three datasets have shown the good performance of the algorithm in comparison to other state-of-the-art algorithms. Additionally, in terms of convergence speed, no performance is lost compared to NMF. Further interesting work would be to add the locality preserving property to the DNMF. We also explored the usage of relative attributes as another format of semantic information in the process of matrix factorization. We showed how this information can be used in

generating discriminative features. At the end, we proposed a new dictionary learning algorithm that uses relative attributes in generating sparse representation of the input signal while enhancing the discriminative property of the signal.



### 3

---

## Immersive Visualization of Image Collections and Feature Space

The world is dealing with a massive amount of collected data, where much of it is visual data (e.g., images and videos). Facebook reports six billion photo uploads per month. The amount of Earth Observation (EO) images is on the order of several terabytes per day. Therefore, browsing and visualizing visual data could help to design new Visual Data Mining (VDM) systems. In such systems, the content of each image (e.g., color, texture, shape) is represented by high-dimensional feature vectors [Low04; SGS10], where the similarity relationship between images is measured based on the distance between feature points. In VDM, a query image is usually presented to the system and the resulting similar images are visualized as thumbnails in a 2D or 3D display space. In interactive VDM [Tal+09; Fog+08; Pan+11], the interface between human and machine plays a key role in enhancing the performance of the system. The interface could provide the user, the ability to quickly grasp the information structure through visualization.

In this chapter, a data visualization system is introduced in order to present visual data (images) and their various aspects, such as feature space, neighborhood graph and tree. Dimensionality Reduction (DR) techniques are used to convert the high-dimensional features into 2D/3D. However, the quality of these techniques must be addressed. Therefore, we describe a novel approach to assess the quality of DR techniques that has appeared in [BDR13]. Since visualization of image collections using DR techniques leads, in most cases, to poor visibility of images, we introduce a customized DR technique based on Non-negative Matrix Factorization (NMF) that accounts for the minimum occlusion and preserves structure. This technique will appear in the *Elsevier Journal of Neurocomputing* [Bab+arb]. In summary, the main contributions of this chapter are:

- introducing immersive data visualization based on virtual reality technology;
- proposing a novel approach to assess the quality of DR techniques;

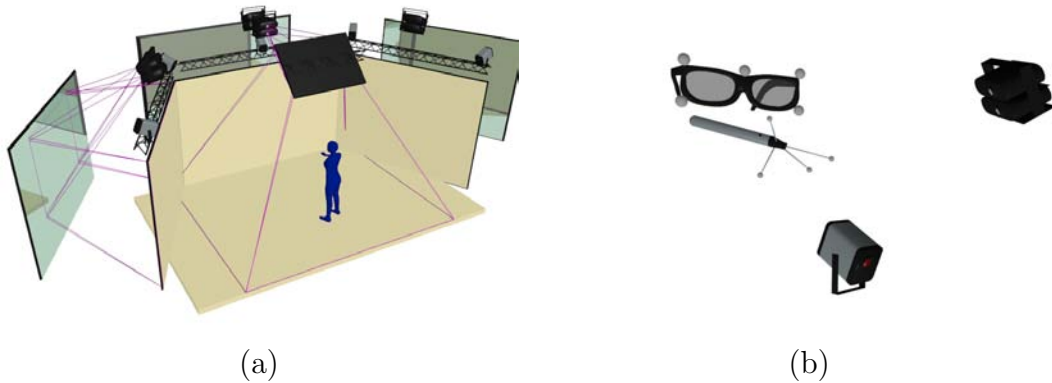


Figure 3.1: (a) A schematic of the CAVE. (b) Cave's devices (Infrared camera, 3D glasses, and projectors)

- and introducing a customized dimensionality reduction technique to arrange image collections in 2D/3D.

In the following, we discuss each aforementioned contribution in detail.

## 3.1 The Cave Automatic Virtual Environment

Immersive Virtual Reality (IVR) is a technology that enables the user to immerse him or herself in a vivid, life-like 3D environment, move in a virtual world and interact with virtual objects. A Cave Automatic Virtual Environment (CAVE) is an immersive virtual reality environment created by projecting a virtual scene to four wall-size screens with stereoscopic projectors behind them. The projectors need to provide a high resolution scene since the user has a very close distance and thus a small pixel size will create the illusion of reality. The projectors work collaboratively with each other to create the virtual world around the user. The user wears a pair of shutter 3D glasses to view the graphics generated by the CAVE. The objects in the CAVE appear to be floating in the air, meaning that the user has the freedom to see the objects from different distances, in different angles, and even from inside of the object. The whole CAVE will bring the user the feeling of reality. A schematic of the CAVE and its objects is presented in Figure 3.1.

### 3.1.1 The CAVE's components

Physically, the CAVE consists of four room-sized walls, projectors, infrared cameras mounted above the walls, and a PC cluster. The projectors/mirrors are located behind the walls and directly/indirectly project onto the walls. The tracking system

is necessary to track the user's movement, like distance to an object, angle of view, and the position of objects. Additionally, the tracking system sends the data to the projection system and the projection system adjusts the view to match the current position of the user. These two systems work cooperatively to create a vivid world in the CAVE.

### 3.1.1.1 PC cluster

The CAVE utilizes a three-layer PC cluster for rendering and visualizing a virtual scene. The first layer is responsible for capturing the user's motions and navigation signals and sends them to the middle layer. Motion capturing is performed by the tracking system and the navigation signals are generated by a Wii controller (Xbox controller). Besides motion capturing, the pose (location and orientation) of the controller is also computed by the tracking system. The middle layer comprises a master PC which is responsible for generating the virtual scene based on the incoming signals from the first layer. This PC first designs the scene and sends a copy of it to each PC in the third layer. Rendering and displaying the scene on the walls is carried out by four PCs (one for each wall). Each PC renders part of the scene and outputs to the corresponding projectors. All PCs are connected via a 1GB Ethernet network to exchange the information. The hardware structure of the CAVE is depicted in Figure 3.2 [BDR13; BRD13a].

### 3.1.1.2 Projection system

The projection system consists of four room-size walls, four pairs of projectors and four PCs. Each pair of projectors consists of two projectors, one for the left eye and one for the right eye, in order to provide a stereoscopic scene. Each pair of projectors handles the scene of one wall and all projectors collaboratively present a whole view for the user who stands inside the CAVE. The two projectors (one emits purple light and the other one green light) project the scene with a small distance between them. Normally, the distance is 0.065m, which is equal to the eye distance of a human being. Thus, a pair of 3D shutter glasses shown in Figure 3.3(a) is needed to see the 3D scene.

### 3.1.1.3 Tracking

The tracking system consists of six infrared cameras mounted above the walls, one control software, and one communication software. Basically, when the user moves his/her head (or position), the scene should be changed accordingly. To simulate this scenario, the tracking system tracks the user's position and movement by computing the positions of markers attached to the glasses. The six infrared cameras are able to detect the markers in their images with roughly 1mm accuracy. The new position

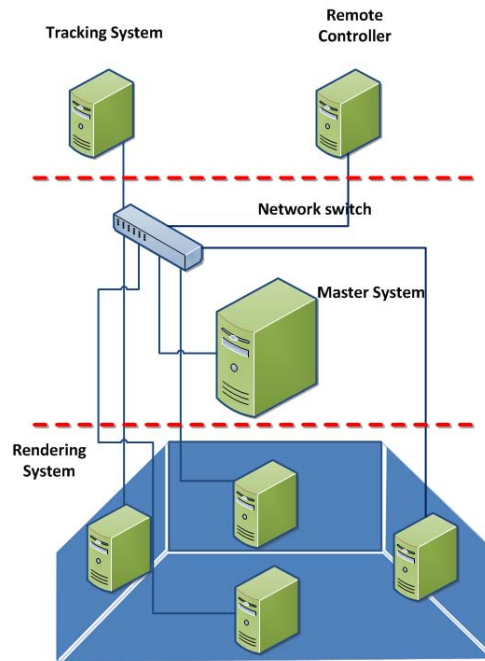


Figure 3.2: The physical diagram of immersive virtual reality. The visualization system consists of three layers with different responsibility. The first layer comprises motion capture (tracking) system and control capturing. A master PC in the middle layer for the synchronization, and finally four systems for rendering for each wall of the CAVE. All systems are connected via an Ethernet network.

of the user or objects are sent to the master PC and this PC reads these positions and arranges the scene accordingly.

#### 3.1.1.4 Software

The visualization software used in the CAVE is called *3DVia Studio Pro* developed by *Dassault Systemes*. It is offered with a Software Development Kit (SDK) to create interactive 3D applications in the Windows operating system. In addition, we use several external libraries for dimensionality reduction, classification (e.g, Support Vector Machines (SVM)), and clustering. Additionally, we also created our own software packages to compute k-means clustering, NMF algorithms, etc.

#### 3.1.1.5 Control system

The control system allows the user to do some actions in the CAVE, like movement, rotation and interaction with objects. The user is provided with an Xbox 360 controller with markers attached. A virtual wand is always displayed which is controlled by this controller. This means that by moving the controller, the direction





Figure 3.3: (a) A pair of shutter glasses with markers attached for tracking the user. (b) An Xbox 360 controller attached with markers to control the scene.

of the wand is also changing accordingly. Moreover, the user can press its buttons to navigate inside the virtual environment. An image of the controller is presented in Figure 3.3(b).

## 3.2 Data visualization

The CAVE is a suitable tool for interactive 3D data visualization, which is widely used in many scientific data visualizations. It allows movement in four directions (e.g., left, right, forward, or backward) and 180 degree horizontal rotation. This allows the user to move towards the desired position by simply changing the orientation of the controller. Zooming is another capability of the system. It is performed by changing the orientation of the wand when it is directed at the point of interest on the screen. Figure 3.4(c) and Figure 3.4(d) show zoom-out and zoom-in modes, respectively.

In any interactive learning technique, the user should be able to interact with and select the visualized data. Here, the user is allowed to select groups of feature points or their corresponding images, which makes the manipulation of the data points more convenient. Using this capability, the user can apply a modification to a number of data points. The proposed group selection tool is a semi-transparent 3D sphere controlled by the controller. The user selects the desired points in the sphere hull by changing the radius and moving the sphere. In order to assist the user, the number of selected items are shown as text to the user. Figure 3.4(a) and Figure 3.4(b) show how the group selection tool performs.

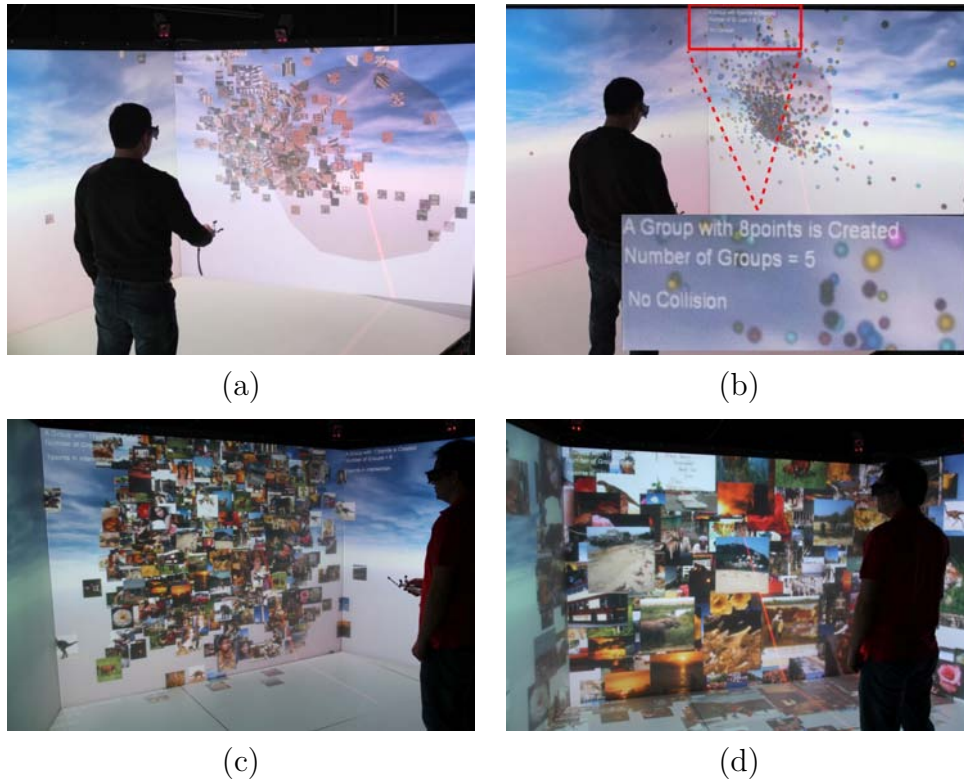


Figure 3.4: (a) The process of selecting a group of data points using a semi-transparent 3D sphere; (b) the number of selected feature points is shown to the user. (c,d) Navigating and exploring images in the CAVE by zooming in and out.

#### 3.2.1 Neighborhood graph and tree

Neighborhood graph and Minimum Spanning Tree (MST) visualization helps to understand the structure of the data. Hence, the proposed system has a graph visualization block that receives a data matrix and then visualizes its neighborhood graph or minimum spanning tree. Two samples of this visualization are depicted in Figure 3.5. In Figure 3.5(a), the neighborhood graph of real data is depicted and in Figure 3.5(b), the MST of synthetic data is presented.

#### 3.2.2 Feature space

Visualization of high-dimensional data (e.g., images) has always been a challenging problem in the area of information mining and visualization. Perhaps the most common way to tackle this problem is to utilize DR techniques to map high-dimensional data to 2D or 3D for visualization. During the last 20 years, numerous methods, both linear or nonlinear, have been proposed to reduce the dimensionality [MPH09; VH08; PWY14; Pan+13; ZWY15; Pom+14]. The most common linear methods

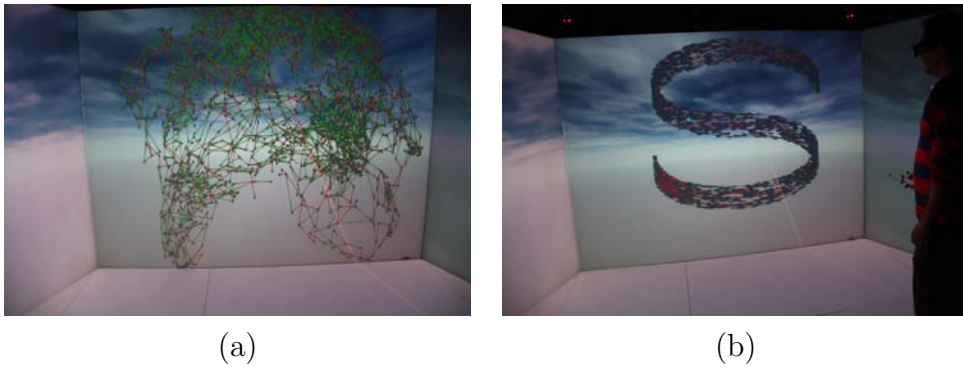


Figure 3.5: Two samples of the visualization of neighborhood graphs (or trees) in the immersive 3D virtual environment. (a) the neighborhood graph of a real data; (b) the Minimum Spanning Tree of a synthetic data set.

are Principal Component Analysis (PCA) [TP91] and Multidimensional Scaling (MDS) [TC10]. Nonlinear methods assume that the data points are coming from a manifold embedded in a high-dimensional ambient space. Depending on whether they preserve the local or global structure of the manifold, they can be categorized, typically, into local and global methods. Local methods like Locally Linear Embedding (LLE) [RS00] and Laplacian Eigenmap (LE) [BN03] emphasize preserving the locality of data points in contrast to global methods like Stochastic Neighbor Embedding (SNE) [HR02a] and Isomap [TDL00], which emphasize preserving the global structure of data points [BRD13a].

In the proposed system, feature space is also visualized by applying different state-of-the-art DR techniques to the original high-dimensional features to map them to 3D. The pipeline of this visualization is depicted in Figure 3.6, where the original features are extracted from the content of an image repository and fed into a dimensionality reduction block to be visualized in the CAVE. The obtained 3D features could also be used to position the images in the CAVE (see Figure 3.7)

### 3.3 Assessment of DR using communication channel model

In the last section, we explained that DR techniques can be used for the visualization of high-dimensional features. However, in the last two decades, dozens of DR techniques have been proposed and show different performance in dealing with different input data sets and tuning parameter(s). Therefore, the question that comes up is if there are any criteria to measure the quality of DR techniques. Since the majority of DR techniques focuses on preserving the local neighborhood distances between data points, state-of-the-art approaches aim at improving their success in

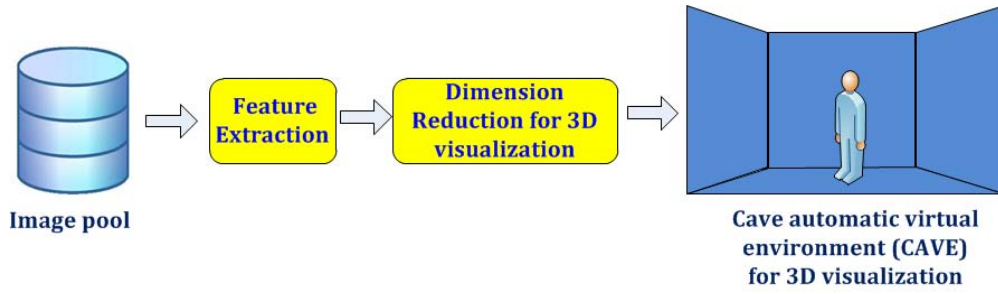


Figure 3.6: An immersive visualization system provides the user with a visual representation of data. Here, high-dimensional features are extracted from a database of Earth Observation images and are fed into a dimensionality reduction technique to be visualized in an immersive 3D virtual environment.

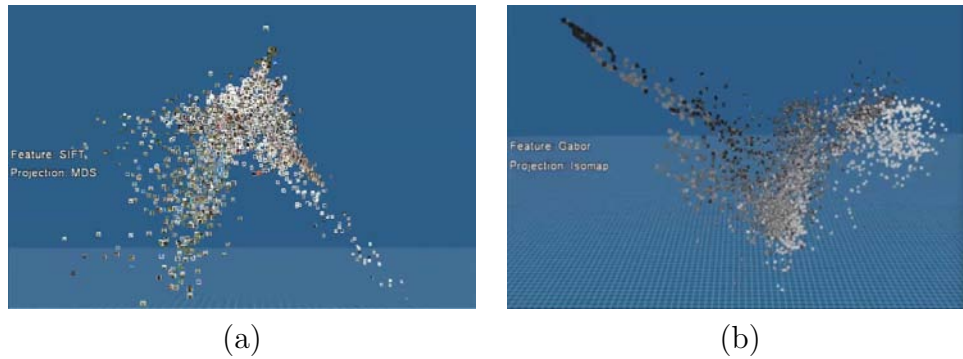


Figure 3.7: Two samples of the visualization of features space of optical and SAR image data sets. (a) optical data set is represented by SIFT feature and MDS performs the dimensionality reduction; (b) the SAR data set is represented by Gabor features and Isomap is DR technique.

preserving the distances. These approaches can be categorized into four categories. The first group evaluates the performance of a technique by assessing the value of the cost function after convergence [Ber+00; BN03]. Clearly, the approaches in this group are useful to compare the results of a few techniques that work based on optimizing a cost function with different parameter(s). The second group focuses on the reconstruction error [BS02]. However, since the reverse transformation does not exist for all techniques, it is hard to employ these approaches for all DR techniques. The third group judges DR techniques based on the classification accuracy applied to the labeled data [MPH09]. The main drawback of this group is the need for labeled data which is not available in most cases. Finally, the last group comprises approaches concentrating on preserving the structure of data. The current criteria for the assessment of the preservation of the data structure are the Local Continuity Meta-Criterion (LCMC) [CB09], the Trustworthiness and Continuity (T&C) [VK06],

and the Mean Relative Rank Error (MRRE) [LV07; LV09]. All these criteria analyze the neighborhoods before and after the dimensionality reduction. Recent work has put all these criteria into a single framework to compare them [LV09]. The advantage of this framework is its ability to propose new criteria for the assessment of DR techniques.

### 3.3.1 Communication channel model

Modeling the information transmission in a processing pipeline as a communication channel is quite interesting in different research areas [CJ10; BD13]. We also consider dimensionality reduction as a communication channel in which data points from a high-dimensional space are transferred into a low-dimensional space [BDR13]. Thus, measuring the quality of this channel reflects the quality of the used dimension reduction technique. Evidently, knowing the fact that recent approaches in DR attempt to preserve the structure of data during dimensionality reduction, we encode the structure of data in a matrix, the so-called ranking matrix [LV09].

#### 3.3.1.1 Ranking matrix

Let's assume a data point in a high-dimensional space is denoted by  $\mathbf{X}_i = [x_{i1}, \dots, x_{iD}]$  and its correspondence in a low-dimensional space by  $\mathbf{Y}_i = [y_{i1}, \dots, y_{iK}]$ , where  $K \ll D$ . The ranking matrices of the data points before and after dimensionality reduction are  $\mathbf{A}$  and  $\mathbf{B}$ , respectively, whose elements are defined by

$$A_{ij} = | \{ \kappa : \alpha_{i\kappa} < \alpha_{ij} \mid (\alpha_{i\kappa} = \alpha_{ij} \& \kappa < j) \} | \quad (3.1)$$

$$B_{ij} = | \{ \kappa : \beta_{i\kappa} < \beta_{ij} \mid (\beta_{i\kappa} = \beta_{ij} \& \kappa < j) \} | \quad (3.2)$$

where  $| \cdot |$  yields the cardinality of the set. The  $\alpha_{ij}$  and  $\beta_{ij}$  represent the distance between the point  $i$  and the point  $j$  in the high and low dimensional spaces, respectively. The  $(i, j)$ .th element of a ranking matrix tells how many data points are closer to the point  $i$  than the point  $j$ . Due to the change of distances between data points during the dimensionality reduction, the ranking matrix of high-dimensional data points (input ranking matrix) changes to the ranking matrix of low-dimensional points (output ranking matrix). The ranking matrices can be interpreted as 2D images and therefore image similarity measures can be employed to quantify the degree of similarity of the input and output ranking matrices. Inspired by medical image registration, the proposed criteria is the Mutual Information (MI) of the probability distribution defined over the joint histogram of ranking matrices.

### 3.3.1.2 Co-ranking matrix

The joint histogram of input and output ranking matrices, namely, the co-ranking matrix [LV08] is defined by:

$$\mathbf{M} = [m_{kl}]_{1 \leq k, l \leq N-1} \quad (3.3)$$

for  $N$  data points, where

$$m_{kl} = |\{(i, j) : (A_{ij} = k) \& (B_{ij} = l)\}|. \quad (3.4)$$

### 3.3.1.3 Mutual information

Mutual Information (MI) and entropy are two metrics that can be utilized to measure the similarity degree of ranking matrices. To this end, a joint probability distribution, namely  $P(i, j)$ , should be defined over the co-ranking matrix by:

$$P(i, j) = \frac{1}{N-1} M \quad (3.5)$$

Therefore, the entropy is defined by

$$H = - \sum_i \sum_j P(i, j) \log P(i, j) \quad (3.6)$$

and the mutual information is:

$$MI = \sum_i \sum_j P(i, j) \log \frac{P(i, j)}{P(i)P(j)} \quad (3.7)$$

Obviously, when a DR technique completely preserves the structure of data points, both ranking matrices are similar and aligned together. Consequently, the co-ranking matrix would be a diagonal matrix with  $N - 1$  on diagonal values. In this case, the mutual information has its maximum value and the entropy has its minimum value.

## 3.3.2 Experiments

To validate our proposed metrics to measure the quality of dimensionality reduction techniques, we performed an experiment and then evaluated and visualized the output of several dimensionality reduction techniques.

### 3.3.2.1 Data sets

We used two data sets, the first one is the UCMerced-Land-Use data set comprising 2100 images categorized in 21 groups. Each group contains 100 image patches of the size  $256 \times 256$  pixels from aerial photography. These images are collected such that they represent rich variation of scene patterns. The second data set is the Corel image data set. This data set contains 1500 images in 15 different groups, where each group contains 100 images.

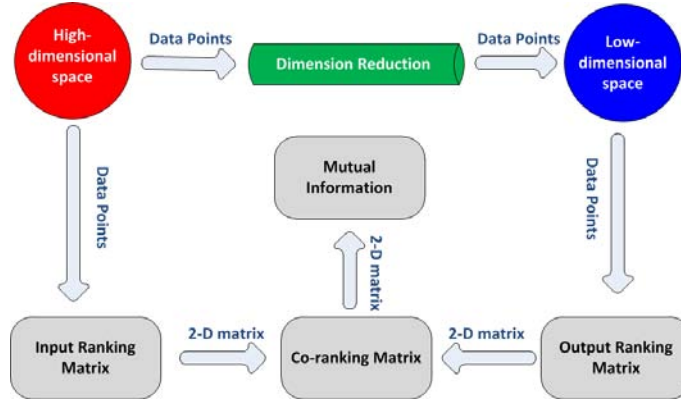


Figure 3.8: The workflow of the proposed approach. While data points are transferred from a high-dimensional space to low-dimensional one, the ranking matrices are built from the data points. These matrices are merged together to build up the co-ranking matrix that is used to define a joint probability distribution. Mutual information computed from this probability distribution is used to assess the quality of dimensionality reduction (here, communication channel) [BDR13].

### 3.3.2.2 Feature extraction

Three different features, namely, color-histogram [SGS10], SIFT [Low04], and Weber Local Descriptor (WLD) [Jie+08; BD13] are extracted from images. The extracted feature descriptors are represented by the Bag-of-Words model, where each image is described by a vector of 200 visual words.

### 3.3.2.3 Dimensionality reduction

We applied three different dimensionality reduction techniques to the high-dimensional features to reduce the dimensionality to 3D for visualization. These techniques are: 1) LE [BN03], 2) SNE [HR02a], and 3) LLE [RS00].

### 3.3.2.4 Results

Mutual information and entropy of the co-ranking matrix are computed for 9 different combinations of features-DR [namely, 1) color-LE, 2) color-SNE, 3) color-LLE, 4) sift-LE, 5) sift-SNE, 6) sift-LLE, 7) WLD-LE, 8) WLD-SNE, 9) WLD-LLE] for both the Merced and the Corel data sets. The computed mutual information and entropy from co-ranking matrices of these combinations are depicted in Figure 3.9.a and Figure 3.9.b for Merced and Corel data sets, respectively. The Figure 3.9.c and Fig. 3.9.d show the 3D plot of the used SNE dimensionality reduction applied to extracted Weber features from the Merced and the Corel data sets, respectively. By looking at the corresponding mutual information (values for number 8 in Figure 3.9.a

and Figure 3.9.b), we conclude that larger mutual information corresponds to better data visualization in terms of seeing better clustering of the data.

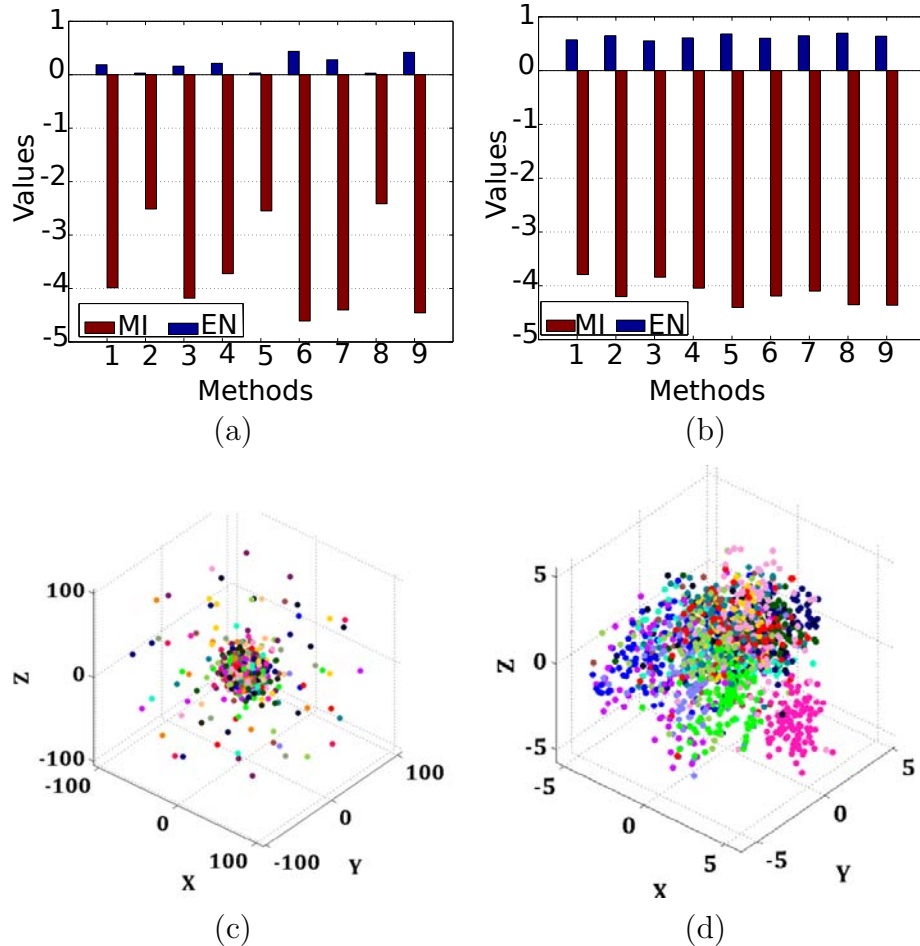


Figure 3.9: The quality of dimensionality reduction applied to the Merced and Corel data sets represented by mutual information and entropy of co-ranking matrix. A combination of three different features (color-histogram, SIFT, and WLD) and three different DR techniques (LE, SNE, LLE) yields 9 feature-DR methods indexed from 1-9; a) results of the Merced data set; b) results of the Corel data set; c) plotted result of method 8 from the Merced data set; d) plotted result of method 8 from the Corel data set.

## 3.4 A customized dimensionality reduction

As mentioned earlier, dimensionality reduction is the most widely employed approach to determine the position of images [BN03; HR02a] in 2D or 3D. However, the main



disadvantage of this approach is that images are usually occluded and much of the display space is not used, giving rise to difficulties for the user to interpret the data. To address this issue, some solutions were proposed to arrange the images in display space based on optimizing a predefined cost function [Wan+10; Mog+04; NW08]. In [Mog+04] and [NW08], the authors estimate the two-dimensional locations of the images by minimizing the overlap between them, where the size of images and display screen are the parameters of the optimization function. They define a cost function as a compromise between similarity preserving and overlap minimization and use the gradient descent method to optimize this cost function. Additionally, the authors in [Wan+10] propose an algorithm that spreads images equally in a given display area, something achieved by minimizing a cost function, which consists of a structure-preserving term, an entropy term, and a term that penalizes locations of images outside the predefined display layout. All the aforementioned methods first reduce the dimensionality of images and then change the position of the data points to fulfill the other requirements. In summary, a good visualization of images should fulfill the three main requirements listed below [NW08]:

- (i) Structure preservation: the relations between images, mainly similarity and dissimilarity, should be preserved;
- (ii) Visibility: all displayed images should be visible to the user (i.e. less overlap between images);
- (iii) Overview: the user should be able to gain an overview of the distribution of images as a cluster.

This section proposes a customized dimensionality reduction technique to arrange image collections in 2D/3D display spaces for image data mining. The main contribution is the development of a novel regularized NMF to position image collections by taking into account the three above requirements. Since each image is represented by a histogram (i.e, Bag-of-Words), there is no conflict in the non-negativity constraint of NMF [FP05], which has non-negative values. In the Bag-of-Words model, each image is treated as a document and its local features as words. The extracted features from all images are pooled and clustered. Next, a histogram of extracted local features from each image is constructed based on cluster centers to represent that image. To consider the aforementioned requirements, a regularization term for each requirement is introduced, which controls the trade-off among requirements. More specifically, there is one regularizer for the structure-preserving requirement, one for the overview requirement and one for the visibility (occlusion-minimizing) requirement. For the structure preserving, the sum of locality (similarity-preserving) [Cai+11] and fairness-preserving [Bab+14c] values are introduced. The Renyi entropy is employed to define the visibility regularizer. Finally, the result of clustering, obtained by applying the k-means algorithm to the original features, is selected to define the overview regularizer. These three regularizers, controlled by some parameters are coupled to the

main objective function of NMF to formulate the cost function of our dimensionality reduction for image positioning. The results of the proposed algorithm are visualized in the CAVE.

### 3.4.1 Regularized NMF for visualization

In order to achieve a good visualization, we require the following constraints for the low-dimensional representation:

#### 3.4.1.1 Structure preservation

An optimal dimensionality reduction technique should preserve the structure of data. In other words, similar images should be placed close to each other and dissimilar images should be far away from each other. To this end, a similarity preserving constraint, which was introduced in the Graph Regularized Non-negative Matrix Factorization (GNMF) algorithm [Cai+11], forces images whose corresponding features are close to each other to also be close to each other in the low-dimensional space. This constraint is defined based on a weight matrix  $\mathbf{W}$ , which represents the internal structure of the high-dimensional data. This matrix is based on the construction of a nearest neighbor graph, where for each point  $\mathbf{x}_j$  we find its  $k$  nearest neighbors and put an edge between  $\mathbf{x}_j$  and each neighbor. Based on this graph, there are many possibilities to construct the matrix  $\mathbf{W}$ . Here we adopt the heat kernel weighting, where:

$$w_{jl} = e^{-\frac{|\mathbf{x}_j - \mathbf{x}_l|^2}{\sigma}}; \text{ subject to } \sigma > 0, \quad (3.8)$$

if nodes  $j$  and  $l$  are connected and 0 otherwise. Based on  $\mathbf{W}$ , the authors of [Cai+11] introduce the following term for similarity preservation in the proposed algorithm which is:

$$\begin{aligned} O_s &= \frac{1}{2} \sum_{j,l} \|\mathbf{v}_j - \mathbf{v}_l\|^2 w_{jl} \\ &= \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}), \end{aligned} \quad (3.9)$$

where  $\mathbf{L} = \mathbf{D} - \mathbf{W}$  and  $\mathbf{D}$  is a diagonal matrix, whose entries are column sums of  $\mathbf{W}$  (i.e.,  $D_{jj} = \sum_l w_{jl}$ ) and  $\mathbf{V}$  is the new data representation.

The constraint for farness, which was introduced in [Bab+14c], forces dissimilar images to remain far away from each other. For this constraint, [Bab+14c] uses the complementary matrix  $\mathbf{W}^{(f)}$  of  $\mathbf{W}$ , which is also based on a graph, where each point  $\mathbf{x}_j$  is connected with its  $p$  farthest neighbors. Based on this graph,  $\mathbf{W}^{(f)}$  is computed by means of binary weighting with

$$w_{jl}^{(f)} = 1, \quad (3.10)$$

if nodes  $j$  and  $l$  are connected and 0 otherwise. Based on  $\mathbf{W}^{(f)}$ , the authors of [Bab+14c] introduce the following term for fairness preservation:

$$\begin{aligned} O_f &= \exp \left[ -\frac{\beta}{2} \sum_{j,l} \|\mathbf{v}_j - \mathbf{v}_l\|^2 w_{jl}^{(f)} \right] \\ &= \exp \left[ -\beta \text{Tr} \left( \mathbf{V}^T \mathbf{L}^{(f)} \mathbf{V} \right) \right], \end{aligned} \quad (3.11)$$

where  $\mathbf{L}^{(f)} = \mathbf{D}^{(f)} - \mathbf{W}^{(f)}$  and  $\mathbf{D}^{(f)}$  is a diagonal matrix, whose entries are column sums of  $\mathbf{W}^{(f)}$ ,  $D_{jj}^{(f)} = \sum_l w_{jl}^{(f)}$ . Parameter  $\beta$  controls how much the fairness property should contribute in general. Combining these two terms leads to the structure-preserving regularization term:

$$O_1 = \lambda_1 \left\{ \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \exp \left[ -\beta \text{Tr} \left( \mathbf{V}^T \mathbf{L}^{(f)} \mathbf{V} \right) \right] \right\}. \quad (3.12)$$

### 3.4.1.2 Occlusion minimization

To minimize the overlap among images and increase the visibility, we propose an entropy term to be coupled with the main NMF objective function. The entropy term is the Renyi quadratic entropy measure [Ren61] of the Gaussian mixture of image positions. We use Renyi entropy because it can be effectively estimated as the sum of pair-wise distances between Gaussian components. It was previously defined in [Wan+10] as

$$H = -\log \left( \frac{1}{N^2} \sum_{i,j} G_{ij} \right) \quad (3.13)$$

with

$$G_{ij} = \exp \left( -\frac{1}{2\sigma^2} |\mathbf{v}_i - \mathbf{v}_j|^2 \right). \quad (3.14)$$

Based on this entropy term, we obtain the following term:

$$O_2 = \lambda_2 \log \left( \frac{1}{N^2} \sum_{i,j} G_{ij} \right). \quad (3.15)$$

### 3.4.1.3 Overview

For the overview, we require points that are in the same cluster in the high-dimensional representation to also be in the same cluster in the low-dimensional representation. To fulfill this requirement, we cluster the points in the high-dimensional space and introduce a graph where each point  $\mathbf{x}_j$  is connected with points belonging to the same cluster. Based on this graph, we define the matrix  $\mathbf{W}^{(o)}$  as

$$w_{jl}^{(o)} = 1, \quad (3.16)$$

if nodes  $j$  and  $l$  are connected and 0 otherwise. Eventually, we introduce the following term for overview:

$$O_3 = \frac{\lambda_3}{2} \sum_{j,l} \|\mathbf{v}_j - \mathbf{v}_l\|^2 w_{jl}^{(o)} = \lambda_3 \text{Tr}(\mathbf{V}^T \mathbf{L}^{(o)} \mathbf{V}), \quad (3.17)$$

where  $\mathbf{L}^{(o)} = \mathbf{D}^{(o)} - \mathbf{W}^{(o)}$  and  $\mathbf{D}^{(o)}$  is a diagonal matrix whose entries are column sums of  $\mathbf{W}^{(o)}$  and  $D_{jj}^{(o)} = \sum_l w_{jl}^{(o)}$ . As we can see, the objective for the overview (3.17) has the same form as the objective for similarity preservation (3.9). Therefore, we combine the two terms into

$$O = \lambda_1 \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \lambda_3 \text{Tr}(\mathbf{V}^T \mathbf{L}^{(o)} \mathbf{V}) = \lambda_1 \text{Tr}(\mathbf{V}^T \tilde{\mathbf{L}} \mathbf{V}) \quad (3.18)$$

where

$$\tilde{\mathbf{L}} = \mathbf{L} + \frac{\lambda_3}{\lambda_1} \mathbf{L}^{(o)}. \quad (3.19)$$

#### 3.4.1.4 Resulting NMF formulation

Adding the introduced terms to the main objective function of NMF leads to the following objective function:

$$O = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 + \lambda_1 \text{Tr}(\mathbf{V}^T \tilde{\mathbf{L}} \mathbf{V}) + \lambda_1 \exp[-\beta \text{Tr}(\mathbf{V}^T \mathbf{L}^{(f)} \mathbf{V})] + \lambda_2 \log\left(\frac{1}{N^2} \sum_{i,j} G_{ij}\right). \quad (3.20)$$

In order to obtain the update rules for  $\mathbf{U}$  and  $\mathbf{V}$ , we expand this objective to

$$O = \text{Tr}(\mathbf{X}\mathbf{X}^T) - 2\text{Tr}(\mathbf{X}\mathbf{V}\mathbf{U}^T) + \text{Tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) + \lambda_1 \text{Tr}(\mathbf{V}^T \tilde{\mathbf{L}} \mathbf{V}) + \lambda_1 \exp[-\beta \text{Tr}(\mathbf{V}^T \mathbf{L}^{(f)} \mathbf{V})] + \lambda_2 \log\left(\frac{1}{N^2} \sum_{i,j} G_{ij}\right) \quad (3.21)$$

and introduce Lagrange multipliers  $\Phi = [\phi_{ik}]$  and  $\Psi = [\psi_{jk}]$  for the constraints  $[u_{ik}] \geq 0$  and  $[v_{jk}] \geq 0$ , respectively. This leads to the Lagrangian

$$\mathcal{L} = \text{Tr}(\mathbf{X}\mathbf{X}^T) - 2\text{Tr}(\mathbf{X}\mathbf{V}\mathbf{U}^T) + \text{Tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) + \lambda_1 \text{Tr}(\mathbf{V}^T \tilde{\mathbf{L}} \mathbf{V}) + \lambda_1 \exp[-\beta \text{Tr}(\mathbf{V}^T \mathbf{L}^{(f)} \mathbf{V})] + \lambda_2 \log\left(\frac{1}{N^2} \sum_{i,j} G_{ij}\right) + \text{Tr}(\Phi\mathbf{U}) + \text{Tr}(\Psi\mathbf{V}). \quad (3.22)$$

The partial derivatives of  $\mathbf{L}$  with respect to  $\mathbf{U}$  and  $\mathbf{V}$  are:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U}_{ik}} = -2(\mathbf{X}\mathbf{V})_{ik} + 2(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik} + \Phi_{ik} \quad (3.23)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{V}_{jk}} &= (-2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U})_{jk} + \lambda_1 \left( 2\tilde{\mathbf{L}}\mathbf{V} \right)_{jk} \\ &\quad - \lambda_1 2\beta \exp \left[ -\beta \text{Tr} \left( \mathbf{V}^T \mathbf{L}^{(f)} \mathbf{V} \right) \right] \left( \mathbf{L}^{(f)} \mathbf{V} \right)_{jk} \\ &\quad + \lambda_2 \frac{2}{\sigma^2 \phi} \sum_l \left[ \mathbf{G}_{lj} (v_{lk} - v_{jk}) \right] + \Psi_{jk} \end{aligned} \quad (3.24)$$

where

$$\phi = \sum_{i,j} \mathbf{G}_{ij}. \quad (3.25)$$

Using the KKT-conditions  $\phi_{ik} u_{ik} = 0$ ,  $\psi_{jk} v_{jk} = 0$  [BV09], we arrive at the following update rules for  $U$  and  $V$ :

$$u_{ik} \leftarrow u_{ik} \frac{(\mathbf{X}\mathbf{V})_{ik}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}} \quad (3.26)$$

$$v_{jk} \leftarrow v_{jk} \mathbf{M} \quad (3.27)$$

and

$$\mathbf{M} = \frac{(\mathbf{X}^T\mathbf{U})_{jk} + \lambda_1 (\tilde{\mathbf{L}}^- \mathbf{V})_{jk} + \lambda_1 \beta R^{(f)} (\mathbf{L}^{(f)+} \mathbf{V}^t)_{jk} + \frac{\lambda_2}{\sigma^2 \phi} \sum_l (\mathbf{G}_{lj} v_{jk}^t) (\mathbf{V} - \mathbf{V}^t)_{jk}^2}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{jk} + \lambda_1 (\tilde{\mathbf{L}}^+ \mathbf{V})_{jk} + \lambda_1 \beta R^{(f)} (\mathbf{L}^{(f)-} \mathbf{V}^t)_{jk} + \frac{\lambda_2}{\sigma^2 \phi} \sum_l (\mathbf{G}_{lj} v_{lk}^t) (\mathbf{V} - \mathbf{V}^t)_{jk}^2} \quad (3.28)$$

where we introduce the following terms:

$$\mathbf{L}^{(f)} = \mathbf{L}^{(f)+} - \mathbf{L}^{(f)-} \quad \text{with} \quad \mathbf{L}_{ij}^{(f)+} = (|\mathbf{L}_{ij}^{(f)}| + \mathbf{L}_{ij}^{(f)})/2, \quad \mathbf{L}_{ij}^{(f)-} = (|\mathbf{L}_{ij}^{(f)}| - \mathbf{L}_{ij}^{(f)})/2, \quad (3.29)$$

$$\tilde{\mathbf{L}} = \tilde{\mathbf{L}}^+ - \tilde{\mathbf{L}}^- \quad \text{with} \quad \tilde{\mathbf{L}}_{ij}^+ = (|\tilde{\mathbf{L}}_{ij}| + \tilde{\mathbf{L}}_{ij})/2, \quad \tilde{\mathbf{L}}_{ij}^- = (|\tilde{\mathbf{L}}_{ij}| - \tilde{\mathbf{L}}_{ij})/2, \quad (3.30)$$

$$R^{(f)} = \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right] \quad (3.31)$$

As expected, the update rule for  $\mathbf{U}$  remains the same as in the original algorithm [LS01], since the newly introduced terms in the objective (3.20) depend only on the variable  $\mathbf{V}$ .

The proof of convergence for the update rule for  $V$  can be found in Appendix B.2.



Figure 3.10: Immersive visualization of image collections in the CAVE

## 3.4.2 Experiments

We conducted several experiments to study and evaluate the performance of the proposed dimensionality reduction technique in order to position several image collections in 3D. To this end, we utilized several different image data sets represented by different features.

### 3.4.2.1 Data sets

We performed experiments on three data sets: 1) Caltech; 2) Corel; 3) SAR.

**Caltech:** This data set contains the 10 biggest groups of the Caltech101 data set, which is 3379 RGB-images. SIFT [Low04] descriptors were extracted from these images, then each image is represented by a 128-dimensional vector using the Bag-of-Word (BoW) model.

**Corel:** This data set contains 1500 64x64 pixels RGB-images in 15 different groups. For the experiment, we extracted local SIFT descriptors from each image and by using the BoW model we created a 200-dimensional feature vector to represent each image.

**SAR:** The SAR data set consists of a collection of 3434 160x160 pixels SAR (Synthetic Aperture Radar) images, which are grouped in 15 classes consisting of various factors such as presences of forests, water, roads and urban area density. We represented each image of this data set with a 64-dimensional feature vector computed by applying BoW model to the extracted SIFT descriptors from the images.

### 3.4.2.2 Setup

For the experiments we selected a random subset of 500 images from each data set and ran each NMF-algorithm 10 times with different starting points and picked the best results. We analyzed the algorithm for different combinations of parameters and for cases where only one of the regularization terms is used. We obtained the combination parameters through cross-validation for each data set separately. More precisely, parameters  $\lambda_1$  and  $\lambda_3$  had a value from the range of  $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4, 10^5\}$  and parameter  $\lambda_2$  from a range of  $\{10^{-1}, 1, 10^1, 10^2, 10^3, 10^4, 10^5\}$ . Therefore, we had 847 ( $11 \cdot 11 \cdot 7$ ) different combinations to test. For the parameters  $\beta$  and  $\sigma$ , we chose the values  $\beta = 10^{-4}$  and  $\sigma = 0.9$  for all data sets obtained from cross validation. The other parameters were also found in the same way and set to  $\lambda_1 = 0.1$ ,  $\lambda_2 = 1000$ ,  $\lambda_3 = 0.01$  for the Caltech data set,  $\lambda_1 = 1$ ,  $\lambda_2 = 1000$ ,  $\lambda_3 = 0.01$  for the Corel data set and  $\lambda_1 = 0.4$ ,  $\lambda_2 = 1000$ ,  $\lambda_3 = 0.01$  for the SAR data set. Then for the visualization and convergence plots, we chose either the combination of the parameters or set the parameter of one regularization term to the given value and others to zero.

For the analysis of each regularization term, we set the other parameters to zero, varied the corresponding parameter and computed the structure preserving, overview, and occlusion for each parameter value. For the occlusion, we treated each image as a cube with constant edge size and computed the overlapping volume of all cubes. For structure preservation, we used the equation

$$P_s = \frac{1}{\text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V})} + \exp[\beta \text{Tr}(\mathbf{V}^T \mathbf{L}^f \mathbf{V})]. \quad (3.32)$$

For the overview, we clustered the images before and after applying the algorithm and computed the percentage of images with the same label. We normalized the resulting values to be in the range  $[0,1]$ .

Finally, we analyzed the effect of each regularization term on the clustering accuracy of the resulting image distribution. For these experiments, we used the parameters given before for each regularization term and then applied the algorithm on a random subset of 500 images for each data set, grouped the resulting distribution into clusters and compared the cluster labels with the ground truth labels. For the comparison, we computed the accuracy and the normalized mutual information to assess the clustering. For details on these metrics please see [Bab+14c]. We repeated each experiment 10 times with different subsets of images and computed the mean value and standard deviation.

### 3.4.2.3 Results

The resulting image visualizations are depicted in Figure 3.11, Figure 3.14, and Figure 3.17 for the Caltech, Corel and SAR data sets, respectively. The results show that

increasing the parameters for overview or structure preservation leads to distributions where similar images are placed close to each other, but at the same time the available space is not used optimally and thus the occlusion of images increases. Using a regularization with high entropy causes the images to be spread over the available space, increasing the image visibility, but at the same time losing the structure of the original distribution. Therefore, a combination of all regularization parameters is required to achieve a high image visibility while preserving the structure of data.

In Figure 3.13(a-e), Figure 3.16(a-e), and Figure 3.19(a-e), we analyze the convergence rate of each algorithm for the three data sets. The plots show that the algorithm converges for each regularization term as well as for their combination. While the entropy term leads to a slight decrease in convergence speed, the overall convergence speed is not significantly impacted.

The behavior of the regularization parameters is analyzed in Figure 3.12, Figure 3.15, and Figure 3.18 for the three data sets. The results show that, as expected, the overview and structure preservation do generally increase when the parameters for either are increased, which, however, increases image occlusion. On the other hand, increasing the parameter for entropy in general leads to a decrease in image occlusion, but at the same time loses the structure and overview.

The results for clustering accuracy in Figure 3.13(f), Figure 3.16(f), and Figure 3.19(f) confirm that by introducing the structure or overview regularization, the performance in most cases increases compared to the NMF. However, for the Caltech data set, the performance slightly decreases for the structure term. Increasing the entropy parameter leads to a decrease in clustering accuracy for all data sets. Finally, the combination of structure, overview and entropy still outperforms the NMF algorithm for the Corel and SAR data sets, but not for the Caltech data set.



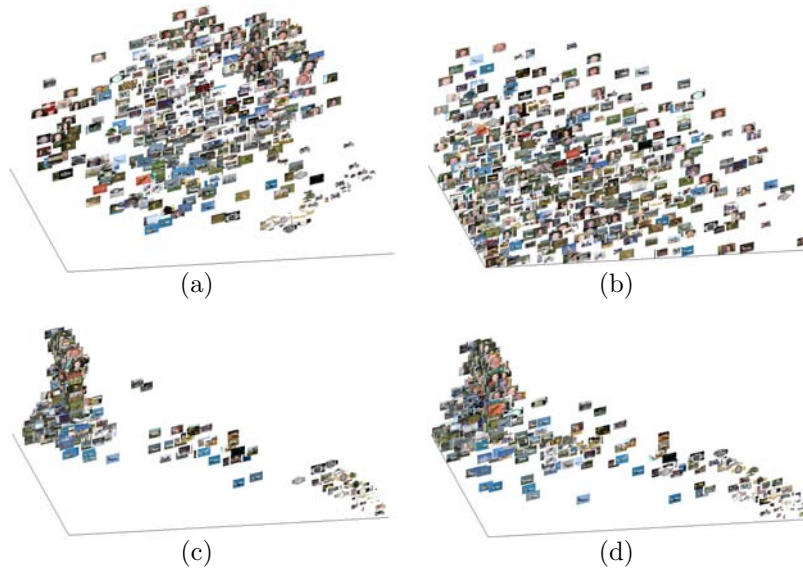


Figure 3.11: Visualizations of the Caltech data set (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF.

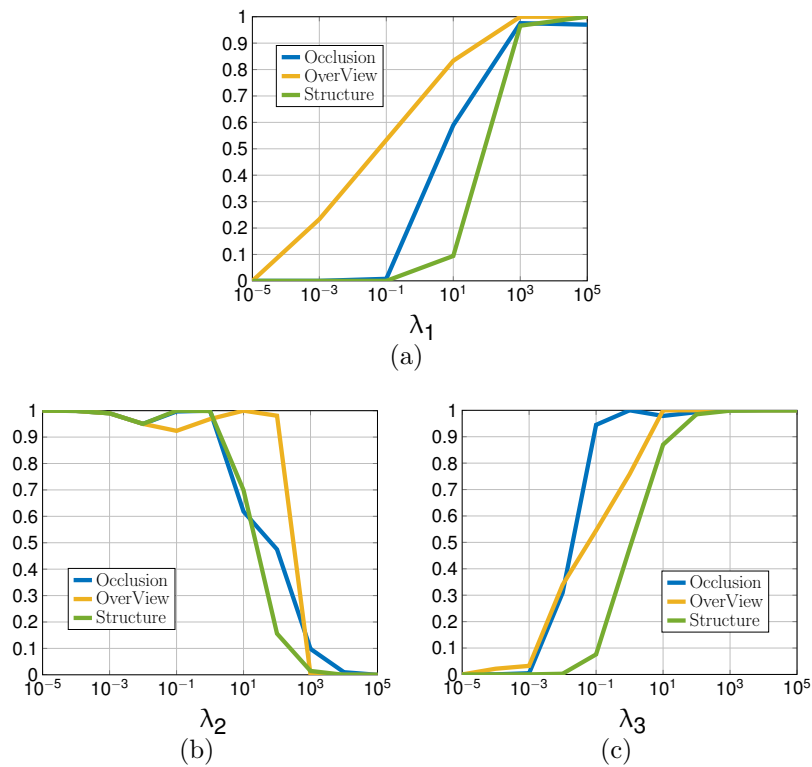


Figure 3.12: Caltech data set analysis of different regularization parameters (a) Structure (b) Entropy (c) Overview

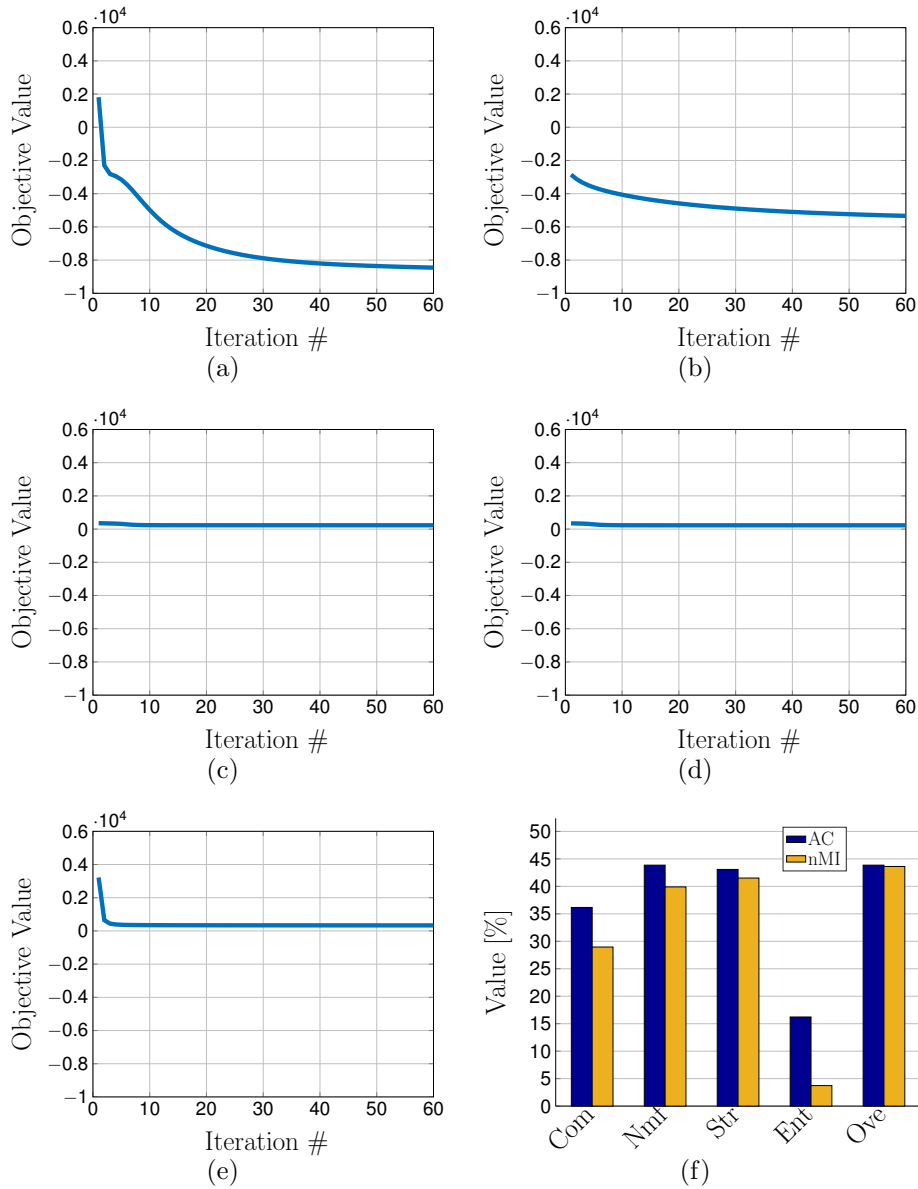


Figure 3.13: Caltech data set convergence plots (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF without regularization (e) Structure regularization (f) Comparison of clustering accuracy for different regularization terms

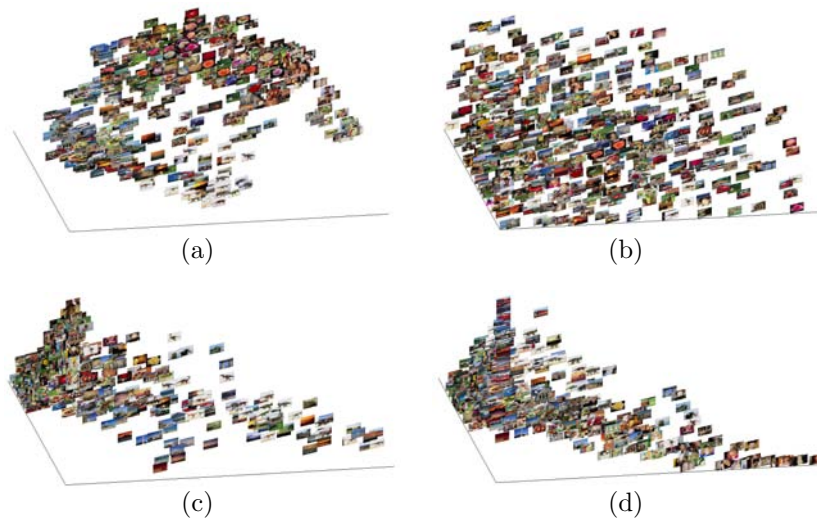


Figure 3.14: Visualizations of the Corel dataset (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF.

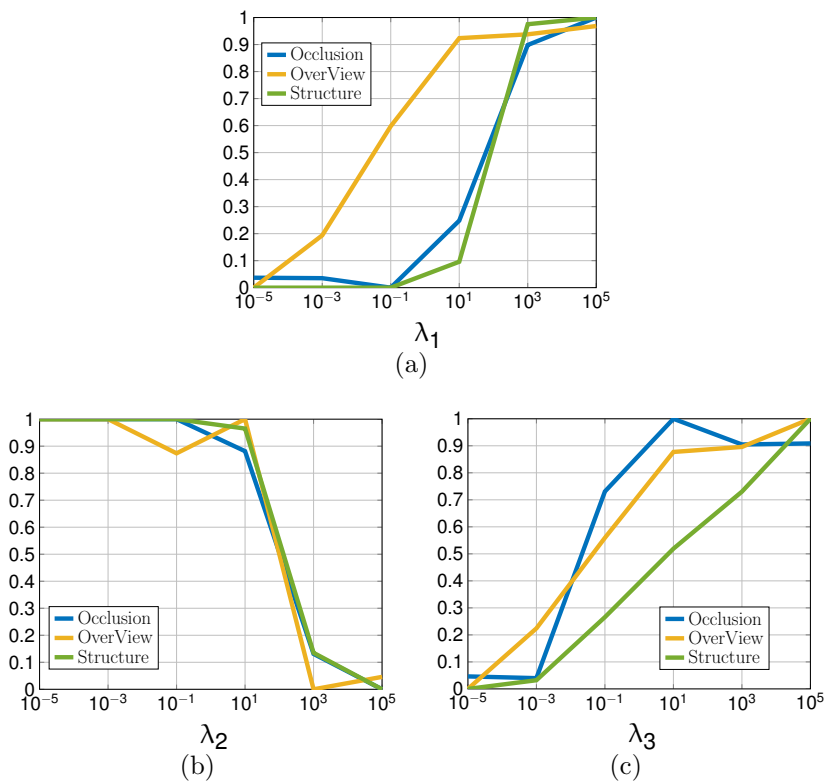


Figure 3.15: Corel dataset analysis of different regularization parameters (a) Structure (b) Entropy (c) Overview

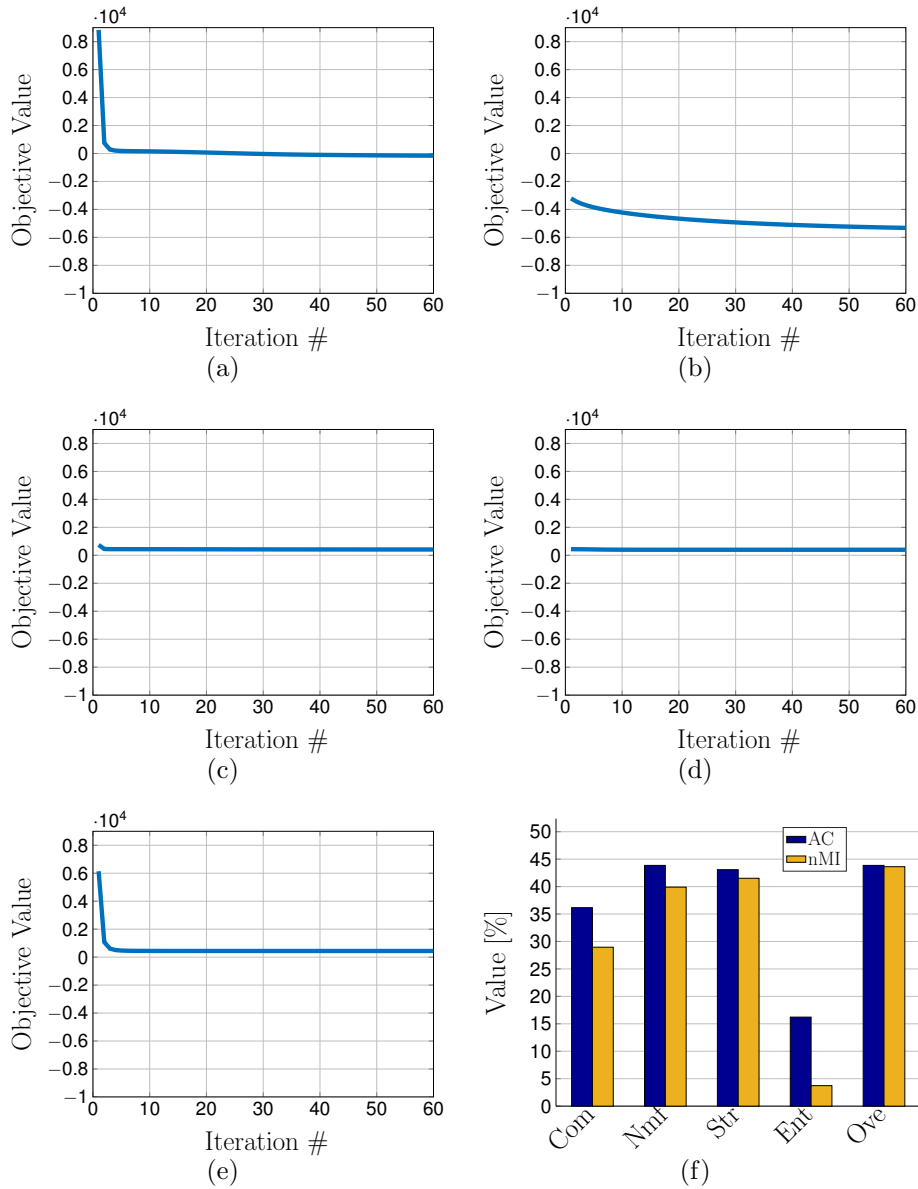


Figure 3.16: Corel dataset convergence plots (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF without regularization (e) Structure regularization (f) Comparison of clustering accuracy for different regularization terms

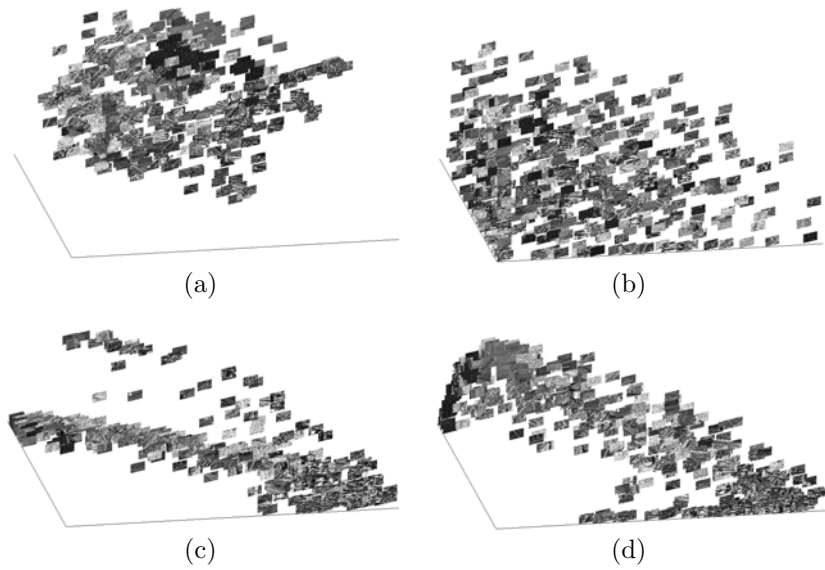


Figure 3.17: Visualization of the SAR dataset (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF.

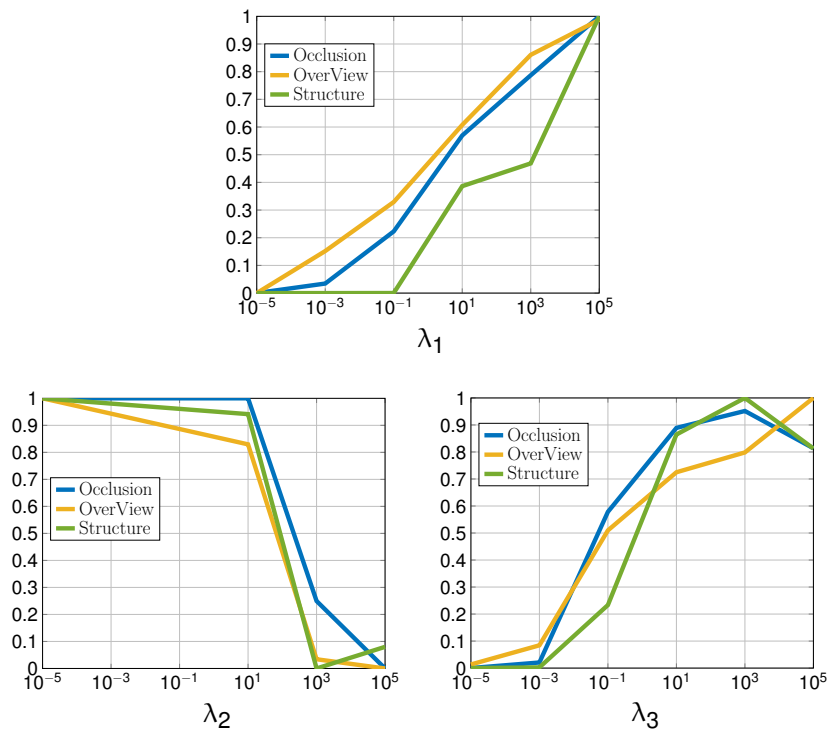


Figure 3.18: SAR dataset analysis of different regularization parameters (a) Structure (b) Entropy (c) Overview

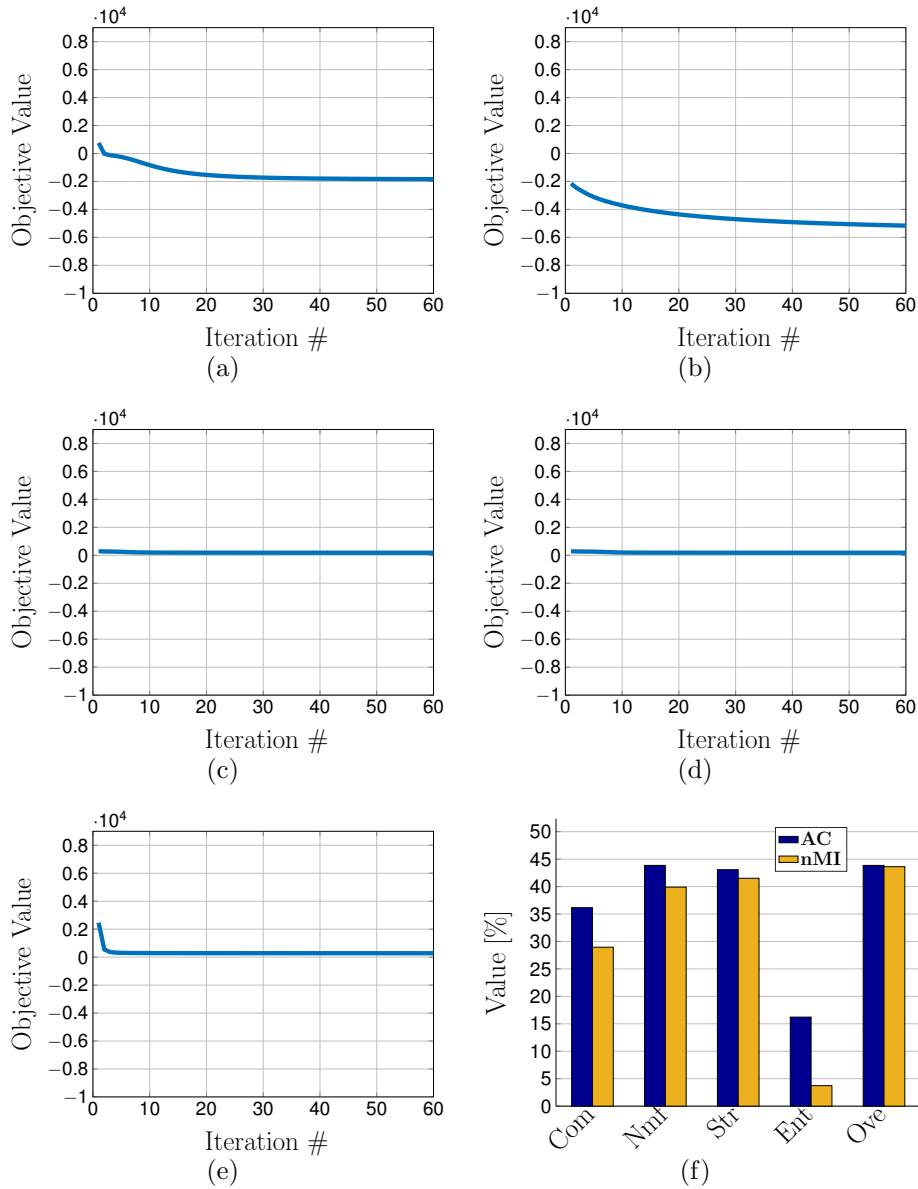


Figure 3.19: SAR dataset convergence plots (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF without regularization (e) Structure regularization (f) Comparison of clustering accuracy for different regularization terms

## 3.5 Summary and conclusion

In this chapter, we focused on the visualization of high-dimensional data. First, we provided detailed information about the CAVE as a 3D virtual environment for interactive data visualization. There, we employed a library of dimensionality reduction to map the data to three dimensions visualization. However, we mentioned that using pure dimensionality reduction might not be a good idea if we have a limited display screen and would like to have less occlusion among images. So, we first proposed a novel evaluation metric to measure the quality of different dimensionality reduction techniques based on the concept of information theory. Then, we proposed a customized dimensionality reduction based on non-negative matrix factorization that takes into account the structure preserving, occlusion, and overview conditions. Specifically, the NMF-algorithm was coupled with proper regularization terms for structure preservation, overview and entropy in order to achieve high visibility, while similar images are still placed close to each other. Experimental results have shown that when a combination of the introduced regularization terms is used, the desired image distribution can be achieved. The update rules for the resulting algorithm have the same form as for the original NMF and convergence is achieved quickly in all cases. One disadvantage of the proposed algorithm is the additional parameters introduced by the regularization terms. Therefore, one possible direction for future work is to reduce the number of parameters by finding the optimal relationship between them.





# Interactive Dimensionality Reduction

Many data analysis techniques deal with massive amounts of data that are represented by very high-dimensional feature vectors. However, the large number of variables in each vector does not necessarily mean that all variables are meaningful and informative. Therefore, it can be useful to obtain a compact and meaningful representation from the content of each data sample. Dimensionality reduction techniques have been widely used to address this problem. However, these techniques are not necessarily devised to generate discriminative low-dimensional features. One way to deal with this issue is to develop interactive techniques, where the user is involved in the process of dimensionality reduction.

In this chapter, we propose several interactive dimensionality reduction techniques based on the framework of non-negative matrix factorization [Bab+14e; Bab+15d]. With an appropriate visualization interface (i.e., the Cave Automatic Virtual Environment (CAVE)), the proposed techniques allow the user to interact with data (images) and provide some constraints to the algorithms. This chapter first begins with an overview of related work and then describes the proposed interactive techniques for dimensionality reduction.

## 4.1 Related work

In this section, we briefly review several related works in the area of interactive dimensionality reduction and data representation.

IN-SPIRE [Wis99] is a well-known visual analytic system for document processing, mainly comprising dimensionality reduction and clustering. It first extracts high-dimensional features from documents utilizing a Bag-of-words model and then applies k-means clustering (with pre-defined number of clusters) to the features for data reduction. In order to visualize these features, Principal Component Analysis

(PCA) is first used to reduce feature dimensionality to just two dimensions before the features are plotted on a screen. Another visual analytic system for document processing is Jigsaw [SGL08], which uses named entities for visualization. In this system, clustering is performed by the k-means algorithm and the results are plotted on a screen. iPCA [Jeo+09] also applies PCA to high-dimensional data for dimensionality reduction. Additionally, it visualizes both low-dimensional data along with the principal axes in high-dimensional space via parallel coordinates. Finally, Testbed [Cho+13] claims to offer an interactive visual system based on a built-in library for dimensionality reduction and clustering. This system aims to help the user to understand data by visualizing the results of different dimensionality reduction and clustering methods. The algorithm claims to reveal valuable knowledge from data and assists the user to choose the most appropriate processing path along with proper parameter(s).

Since the last decade, numerous projects have utilized virtual reality for information visualization. For instance, VRMiner provides a 3D interactive tool for visualizing multimedia data utilizing virtual reality [Azz+05]. In this system, a set of numeric and symbolic attributes, along with multimedia data (e.g., music, video, and websites), are presented in a 3D virtual environment. One drawback of the system, however, is that images and videos need to be displayed on a second PC in order to have a real-time system. Another sample illustrating the usage of virtual reality for information visualization is an application named 3D MARS [NH01]. This application is mainly for content-based image retrieval, in which the user browses and queries images in an immersive 3D virtual environment. The main aim of this system is visualizing the results in 3D space.

Besides the aforementioned technologies for the visualization and exploration of data, Human-Computer Interaction (HCI) has shown valuable contribution in the domain of data mining and knowledge discovery. The main goal is to provide the user with a way to learn how to analyze the data in order to get knowledge to make proper decisions. For example, Holzinger [Hol12] has investigated HCI for interactive visualization of biomedical data. As another example, Wong et al [WXH11] have shown first the similarity between intelligent information analysis and medical diagnosis, and then proposed which aspects should be considered during the design of an interactive information visualization to facilitate intelligent information analysis.

Evidently, the main processing step in every visual analytic system is dimensionality reduction. Since the last decade, numerous linear and nonlinear DR techniques have been proposed in different research areas. While linear approaches assume data lies in a linear  $d$ -dimensional subspace of a high-dimensional feature space, nonlinear approaches consider data as a  $d$ -dimensional manifold embedded in a high-dimensional space. Perhaps the most famous linear algorithm is PCA that projects data into  $d$  eigenvectors corresponding to  $d$  largest eigenvalues of the covariance matrix of the data. Among nonlinear methods, Locally Linear Embedding (LLE) [RS00] aims to preserve the structure of data during dimension reduction. It

assumes that the data belongs to a low-dimensional smooth and nonlinear manifold that is embedded in an ambient high-dimensional space. The data points are mapped to a lower-dimensional space while preserving the neighborhood.

Laplacian Eigenmap (LE) [BN03] is a nonlinear technique in the domain of spectral decomposition methods and locally transforms data into low-dimensional space. It performs this transformation by building a neighborhood graph from the given data, whose nodes represent data points and whose edges depict the proximity of neighboring points. This graph approximates the low-dimensional manifold embedded in a high-dimensional space. The eigen-functions of the Laplace Beltrami operator on the manifold serve as the embedding dimensions.

Stochastic Neighbor Embedding (SNE) [HR02a] is a probabilistic based approach attempting to preserve the neighborhoods of points based on converting the distances into probabilities. Therefore, the neighborhood relation between two data points is represented by a probability such that closer points to a specific point have larger probability than further points. Thereafter, data points are mapped to low-dimensional space such that the computed probabilities are preserved. This is done by minimizing the sum of the Kullback-Leibler Divergence (KLD) of the probabilities.

## 4.2 Immersive data visualization

We have implemented an immersive 3D virtual environment (i.e., CAVE) in order to visualize images and allow the user to interact with them. The 3D positions of images in the CAVE are determined by the clustering result of k-means method and the distances between the images and their corresponding centers. The user can easily navigate inside the data and check the result of k-means clustering. If an image is mis-clustered, the user can correctly connect this image to all its semantically similar images. For convenience, the user can directly connect this image to a target cluster center instead of connecting it individually to all the images of the target cluster. Additionally, the distance between an image and its cluster center is proportional to its distance to the cluster center in the high-dimensional space. Thus, the user can first scan the images that are far away from their centers because they are prone to be mis-clustered. A snapshot of the visualization of images in the CAVE is depicted in Figure 4.1(a) and Figure 4.1(d) for SAR and optical data, respectively. Compared to the approaches that present images for labeling to the user sequentially, the main advantage of the immersive visualization technology is that it provides the user an overview of the whole data set, assisting the user to find those images that should be matched more efficiently. The 3D visualization of images with sample interaction for two different datasets is depicted in Figure 4.1(c), Figure 4.1(d).

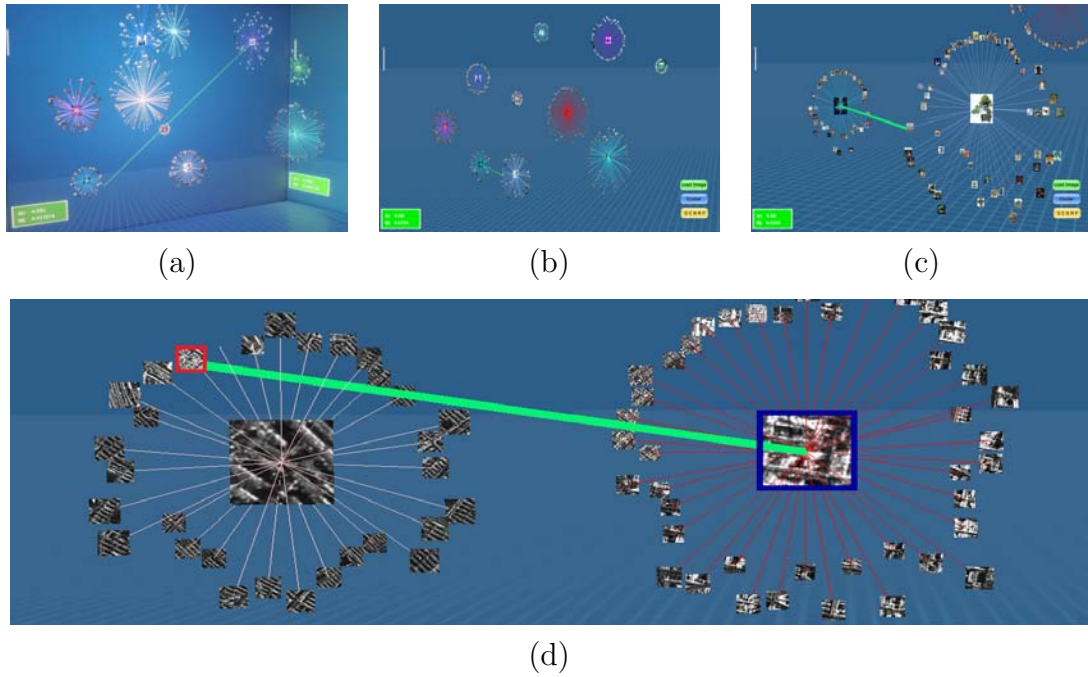


Figure 4.1: (a) the visualization of the clustering result in the CAVE. Images are positioned around their cluster centers based on their distances. A sample image of each cluster is used to depict the cluster center. (b) and (c) show user interactions on a desktop. A mis-clustered image is connected to a semantically cluster center by a green line. (d) a mis-clustered image (the image with red border) is connected (green line) to the cluster center of target cluster (with blue border) for SAR images. This interaction updated the semantic similarity matrix  $W$ , which is used in our novel NMF algorithms.

### 4.3 Interactive algorithms

Presented in this section are several novel interactive Dimensionality Reduction (DR) algorithms based on non-negative matrix factorization. These algorithms are: 1) Variance Regularized Non-negative Matrix Factorization (VNMF); 2) Center-Map Non-negative Matrix Factorization (CMNMF); 3) Pair-wise Constrained NMF; and 4) Set-wise Constrained NMF. The first two algorithms use the same user interface in which images are visualized as a cluster obtained by k-means algorithm. A snapshot of this visualization is presented in Figure 4.1. Images in the user interface of last two algorithms are visualized based on their position computed by a dimensionality reduction. Here, the user chooses a DR technique and the images are displayed in the position of their corresponding 3D features.

### 4.3.1 Variance constrained NMF

The goal of VNMF is to factorize some input data  $\mathbf{X}$  into two non-negative matrices  $\mathbf{U}$  and  $\mathbf{V}$ , subject to a minimum value for the variance of  $\mathbf{V}$  (i.e.,  $\sigma_V^2$ ). More precisely, we minimize the following objective function:

$$\begin{aligned} F &= \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda\sigma_v^2 \\ \text{s.t. } \mathbf{U} &= [u_{ik}] \geq 0, \\ \mathbf{V} &= [v_{jk}] \geq 0 \end{aligned} \quad (4.1)$$

The user interactions are kept in the matrix  $\mathbf{W} \in \mathbb{R}^{N \times N}$ , whose elements  $w_{ij}$  are 1 if the images  $i$  and  $j$  are connected or 0 if they are not. To compute the variance of new features, their expectation value should be computed first. Thus, we scale the matrix  $\mathbf{W}$  so that its rows always sum to 1, yielding matrix  $\widetilde{\mathbf{W}}$ . The multiplication of  $\widetilde{\mathbf{W}}$  and  $\mathbf{V}$  finally results in a matrix  $\overline{\mathbf{V}}$ , holding the mean features of similar images. For example, given a dataset of four images, where image 1 is connected to image 2, we get:

$$\overline{\mathbf{V}} = \widetilde{\mathbf{W}}\mathbf{V} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{pmatrix} \quad (4.2)$$

By introducing new matrix  $\mathbf{T} = \mathbf{I} - \widetilde{\mathbf{W}}$ , we can write

$$\sigma_V^2 = \|\mathbf{V} - \overline{\mathbf{V}}\|_F^2 = \|\mathbf{V} - \widetilde{\mathbf{W}}\mathbf{V}\|_F^2 = \|\mathbf{T}\mathbf{V}\|_F^2. \quad (4.3)$$

In order to control the variance, another scalar parameter  $\theta$  is introduced inside the regularizer. Finally, the objective function to be minimized is

$$\begin{aligned} C &= \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda(N\theta - \|\mathbf{T}\mathbf{V}\|_F^2)^2 \\ &= \sum_i^D \sum_j^N (x_{ij} - \sum_{k=1}^K u_{ik}v_{jk})^2 \\ &\quad + \lambda \left( N\theta - \sum_i^N \sum_j^K (v_{ij} - \bar{v}_{ij})^2 \right)^2. \end{aligned} \quad (4.4)$$

where  $\lambda$  controls the overall contribution of the regularizer.

### 4.3.1.1 Optimization rules

To minimize the cost function given in (4.4), we first expand it to

$$\begin{aligned}
 O &= \text{Tr}\left((\mathbf{X} - \mathbf{UV}^T)(\mathbf{X} - \mathbf{UV}^T)^T\right) \\
 &+ \lambda\left(N\theta - \text{Tr}((\mathbf{TV})(\mathbf{TV})^T)\right)^2 \\
 &= \text{Tr}(\mathbf{XX}^T) - 2\text{Tr}(\mathbf{XVU}^T) + \text{Tr}(\mathbf{UV}^T\mathbf{VU}^T) + \lambda\mathbf{Z}^2,
 \end{aligned} \tag{4.5}$$

where

$$\mathbf{Z} = N\theta - \text{Tr}(\mathbf{TVV}^T\mathbf{T}^T). \tag{4.6}$$

We define Lagrange multipliers  $\alpha_{ik}$  and  $\beta_{jk}$  with the constraints  $u_{ik} \geq 0$  and  $v_{jk} \geq 0$ , respectively. Therefore, by defining  $\Phi = [\alpha_{ik}]$  and  $\Psi = [\beta_{jk}]$ , the Lagrangian  $\mathcal{L}$  is

$$\begin{aligned}
 \mathcal{L} &= \text{Tr}(\mathbf{XX}^T) - 2\text{Tr}(\mathbf{XVU}^T) + \text{Tr}(\mathbf{UV}^T\mathbf{VU}^T) \\
 &+ \lambda\mathbf{Z}^2 + \text{Tr}(\Phi\mathbf{U}) + \text{Tr}(\Psi\mathbf{V}).
 \end{aligned} \tag{4.7}$$

The partial derivatives of  $\mathcal{L}$  with respect to  $\mathbf{U}$  and  $\mathbf{V}$  are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = -2\mathbf{XV} + 2\mathbf{UV}^T\mathbf{V} + \Phi \tag{4.8}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = -2\mathbf{X}^T\mathbf{U} + 2\mathbf{VU}^T\mathbf{U} - 2\lambda\mathbf{ZT}^T\mathbf{TV} + \Psi. \tag{4.9}$$

Using the Karush-Kuhn-Tucker (KKT) conditions [BV09], where  $\alpha_{ij}u_{ij} = 0$  and  $\beta_{jk}v_{jk} = 0$ , the following equations are obtained:

$$-(\mathbf{XV})_{ik}u_{ik} + (\mathbf{UV}^T\mathbf{V})_{ik}u_{ik} = 0 \tag{4.10}$$

$$[-\mathbf{X}^T\mathbf{U} + \mathbf{VU}^T\mathbf{U} - \lambda\mathbf{ZT}^T\mathbf{TV}]_{jk}v_{jk} = 0 \tag{4.11}$$

With the symmetric matrices  $\mathbf{T} = \mathbf{T}^+ - \mathbf{T}^-$ , where  $\mathbf{T}_{ij}^+ = (|\mathbf{T}_{ij}| + \mathbf{T}_{ij})/2$  and  $\mathbf{T}_{ij}^- = (|\mathbf{T}_{ij}| - \mathbf{T}_{ij})/2$ , the update rules for  $\mathbf{U}$  and  $\mathbf{V}$  can be rewritten as:

$$u_{ik} \leftarrow u_{ik} \frac{(\mathbf{XV})_{ik}}{(\mathbf{UV}^T\mathbf{V})_{ik}} \tag{4.12}$$

$$v_{jk} \leftarrow v_{jk} \frac{(\mathbf{X}^T\mathbf{U} - 2\lambda\mathbf{ZT}^+\mathbf{T}^-\mathbf{V})_{jk}}{(\mathbf{VU}^T\mathbf{U} - \lambda\mathbf{ZT}^+\mathbf{T}^+\mathbf{V} - \lambda\mathbf{ZT}^-\mathbf{T}^-\mathbf{V})_{jk}} \tag{4.13}$$

### 4.3.2 Center Map NMF

In CMNMF, the user-interaction/semantic information is incorporated/injected inside the main function of NMF. First, we create a symmetric diagonal matrix  $\mathbf{W}_{N \times N}$  ( $N$  is the number of images) with 1 in the main diagonal that holds the semantic similarity of images. When the user links an image  $i$  to a cluster center  $c$  containing  $p$  images, the corresponding  $p$  elements of the row  $i$  of matrix  $\mathbf{W}$  would be 1. In other words,  $w_{ij} \neq 0$  shows the images  $i$  and  $j$  are semantically similar to each other. Finally, matrix  $\mathbf{W}$  is updated as a weight matrix with  $w_{ij} = \sum_{l=1}^M w_{il} = 1$ , where  $M$  is the total number of non-zero elements in row  $i$ . In addition, we introduce an auxiliary matrix  $\mathbf{Z}$  in order to get matrix  $\mathbf{V}$ . For example, suppose there are four images, the operation is:

$$\begin{aligned} \mathbf{V} = \mathbf{WZ} &= \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ z_{31} & z_{32} & z_{33} & z_{34} \\ z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix} \\ &= \begin{pmatrix} \frac{z_{11} + z_{21}}{2} & \frac{z_{12} + z_{22}}{2} & \frac{z_{13} + z_{23}}{2} & \frac{z_{14} + z_{24}}{2} \\ \frac{z_{11} + z_{21}}{2} & \frac{z_{12} + z_{22}}{2} & \frac{z_{13} + z_{23}}{2} & \frac{z_{14} + z_{24}}{2} \\ z_{31} & z_{32} & z_{33} & z_{34} \\ z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix} \end{aligned} \quad (4.14)$$

From this example, we can see that the new representation guarantees that the first two images (first two rows of  $\mathbf{V}$ ) will be assigned to the same cluster center. Adding the introduced terms to the NMF formulation leads to the following minimization objective:

$$\min_{\mathbf{U}, \mathbf{Z}} \mathcal{O} = \min_{\mathbf{U}, \mathbf{Z}} \|\mathbf{X} - \mathbf{UZ}^T \mathbf{W}^T\| \quad (4.15)$$

where  $\mathbf{X}$ , the original image representations, is decomposed into  $\mathbf{U}$  and  $\mathbf{V}$ ,  $\mathbf{V}$  being the new representation of the images and  $\mathbf{U}$  being the bases. For the derivation of the update rules we expand this objective to

$$\begin{aligned} \mathcal{O} &= \text{Tr}((\mathbf{X} - \mathbf{UZ}^T \mathbf{W}^T)(\mathbf{X} - \mathbf{UZ}^T \mathbf{W}^T)^T) \\ &= \text{Tr}(\mathbf{X}\mathbf{X}^T) - 2\text{Tr}(\mathbf{X}\mathbf{W}\mathbf{Z}\mathbf{U}^T) + \text{Tr}(\mathbf{U}\mathbf{Z}^T \mathbf{W}^T \mathbf{W}\mathbf{Z}\mathbf{U}^T) \end{aligned} \quad (4.16)$$

and introduce Lagrange multipliers  $\Phi = [\phi_{ik}]$  and  $\Psi = [\psi_{jk}]$  for the constraints  $[u_{ik}] \geq 0$  and  $[v_{jk}] \geq 0$ , respectively. This leads to the Lagrangian

$$\mathcal{L} = \mathcal{O} + \text{Tr}(\Phi \mathbf{U}^T) + \text{Tr}(\Psi \mathbf{Z}^T) \quad (4.17)$$

The partial derivatives of  $\mathcal{L}$  with respect to  $\mathbf{U}$  and  $\mathbf{Z}$  are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = -2\mathbf{X}\mathbf{W}\mathbf{Z} + 2\mathbf{U}\mathbf{Z}^T\mathbf{W}^T\mathbf{W}\mathbf{Z} + \Phi \quad (4.18)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = -2\mathbf{W}^T\mathbf{X}^T\mathbf{U} + 2\mathbf{W}^T\mathbf{W}\mathbf{Z}\mathbf{U}^T\mathbf{U} + \Psi \quad (4.19)$$

Using the Karush-Kuhn-Tucker (KKT) conditions [BV09], we arrive at the following update rules for  $\mathbf{U}$  and  $\mathbf{Z}$ :

$$u_{ij} \leftarrow u_{ij} \frac{(\mathbf{X}\mathbf{W}\mathbf{Z})_{ij}}{(\mathbf{U}\mathbf{Z}^T\mathbf{W}^T\mathbf{W}\mathbf{Z})_{ij}} \quad (4.20)$$

$$z_{jk} \leftarrow z_{jk} \frac{(\mathbf{W}^T\mathbf{X}^T\mathbf{U})_{jk}}{(\mathbf{W}^T\mathbf{W}\mathbf{Z}\mathbf{U}^T\mathbf{U})_{jk}} \quad (4.21)$$

### 4.3.3 Immersive interface

We utilized a 3D interactive application development software, namely 3D Via Studio, to create our interactive user interface. The application can be run on a desktop PC or in a CAVE [BRD13a]. The 3D positions of images are determined by the clustering result of the k-means method, as well as the distances between the images and their corresponding centers. More precisely, the distance between an image and its cluster center is proportional to its actual distance in the high-dimensional space. Thus, the user can first assess the images that are far away from their respective cluster centers, because they are prone to be mis-clustered.

In addition to the CAVE, we also implemented a desktop version of our application that ran on a single PC. A snapshot of the immersive visualization of the clustering results in the CAVE is depicted in Figure 4.1(a). Figure 4.1(b) and Figure 4.1(c) depict two views of a clustering result visualized on a desktop PC. The main advantage of using the immersive visualization technology is that the user gets an overview of the whole dataset and can therefore identify those images that should be relabeled more efficiently. This is especially important if the user is dealing with a high amount of images that cannot be visualized on a monitor with limited space.

### 4.3.4 Experiment 1

In this experiment, we evaluate the performance of our two introduced interactive dimensionality reduction algorithms.



#### 4.3.4.1 Data set

The data set used in our experiments is a Synthetic Aperture Radar (SAR) data set [Bab+14c] represented by three different features. It consists of a collection of 3434 SAR images of the size  $160 \times 160$  pixels, pre-categorized into 15 classes/labels (by a SAR expert) based on the presences of forests, water, roads and urban area density. For instance, one image is categorized as “sea” and another as “industrial part”. Three different feature vectors, namely Weber Local Descriptor (WLD) [Jie+08], Mean-Variance [CDD13] and Image Intensity [CDD13] were extracted from the images, leading to a total of 64 dimensions in these three cases. All features are normalized to lie between 0 and 1. The information about this data set is presented in Appendix A.

#### 4.3.4.2 Evaluation metrics

Two evaluation metrics, namely Accuracy (AC) and normalized Mutual Information (nMI) are used to assess the quality of k-means clustering applied on the original and learned representations of images [XLG03].

#### 4.3.4.3 Design

Given a set of  $N$  images, we randomly selected 10 – 15 percent of the images as training data. In our experiment, 500 images are chosen as training data. For this new image set, we applied k-means clustering algorithm and visualized the images by the cluster-based visualization system. We visualized the images by 15 clusters. The user navigated inside the data and corrected the mis-clustered images by drawing a green line between the image and the center of the desired target cluster. The interactions were saved in a matrix with 2 columns and  $I$  rows, where  $I$  is the number of interactions. The first column stored the ID of the interacted image and the second column saved the ID of the target cluster. The interactive matrix was used to create the weighted similarity matrix for systems based on VNMF and CMNMF. For example, the user moved image  $i_2$  to the cluster  $c_2$ . And there were two images  $i_6$  and  $i_7$  belonged to  $c_2$ . The images  $i_2, i_6$  and  $i_7$  would be regarded as similar images and the positions in similarity matrix  $(2, 6), (2, 7), (6, 2), (6, 7), (7, 2)$  and  $(7, 6)$  would be set to 1. After updating the matrix  $\mathbf{W}$ , the matrix  $\widetilde{\mathbf{W}}$  was determined by calculating row-wise average, namely

$$\widetilde{w}_{ij} = \frac{w_{ij}}{\sum_{l=1}^N w_{il}} \quad (4.22)$$

where  $N$  was the number of images in total. Then, for VNMF, the matrix  $\mathbf{T}$  was updated by  $\mathbf{T} = \mathbf{I} - \widetilde{\mathbf{W}}$  where  $\mathbf{I}$  was an identity matrix. The new matrix  $\mathbf{T}$  is used in VNMF to calculate the new data representation  $\mathbf{V}$  where  $\mathbf{V}$  shows better clustering result in low dimensional space. For CMNMF, the matrix  $\widetilde{\mathbf{W}}$  was directly used in

updating rules to obtain the new data representation. All similar images would be mapped to their semantic centers and as a result, the new representations provided good clustering accuracy.

The dimension of new representation matrix  $\mathbf{V}$  was set to the number of images by the number of classes. For our SAR data set, the dimension of  $\mathbf{V}$  was set to be  $N \times 15$ . For CMNMF, the similarity matrix was directly applied to the updating rules since the matrix  $\mathbf{V}$  was replaced by  $\mathbf{WZ}$ . Thus, no parameter was needed. For VNMF, the regularization parameter was used to control the contribution of the user interactions. The parameter was chosen by tuning the value  $\lambda$  by searching the grid  $\{10^{-7}, 10^{-6}, 10^{-5}, \dots, 10^2, 10^3\}$ . Since the new representation  $\mathbf{V}$  consisted of small values ( $0 \sim 0.6$ ), and the updating rules were sensitive to the regularization terms, the parameter between  $10^{-6}$  and  $10^{-2}$  was chosen in most cases.

After training, we used these three learning algorithms in two different ways for test data: 1) cluster the test data following the change of interaction number and 2) cluster the test data following the change of the dimension of new representations.

Traditionally, after training, the test data was processed as a whole data set (batch processing). Since the size of test data set is much larger than the size of training data, divide-and-conquer processing is used when clustering the test data. Compared with batch processing, divide-and-conquer processing provides similar performance in clustering accuracy with much less running time. The test data was divided into parts with the same size of training data. We then mixed the training data with each part and applied learning algorithms to obtain new representations for each part. The clustering results of k-means algorithm on each part were averaged as the final result of the whole data set. A schematic of this process is depicted in Figure 4.2.

When clustering the test data following the change of interaction number, the dimension of the matrix  $\mathbf{V}$  is fixed to the number of images by the number of classes. Based on different numbers of interactions, the test data are classified and the clustering results are calculated.

When clustering the test data following the change of the dimension of new representations, the number of interactions is fixed to 180. The new representation matrix  $\mathbf{V}$  will have the size of  $N \times k$ , where  $N$  is the number of images,  $k$  changes in the range 3, 6, 9, 12, 15 for the SAR data set. For matrix  $\mathbf{V}$  with different sizes, the clustering results are calculated and compared with other algorithms.

To enhance the user's effect, the locality property was used to propagate the user's interaction. In other words, the user's interaction on training data would be applied to their nearest neighbors in test data. The idea is shown in Figure 4.3. The usage of the locality property could produce errors since it uses Euclidean distance to find nearest neighbor. However, it strengthened the user interaction and the learning algorithm would correct the error produced by the locality property.

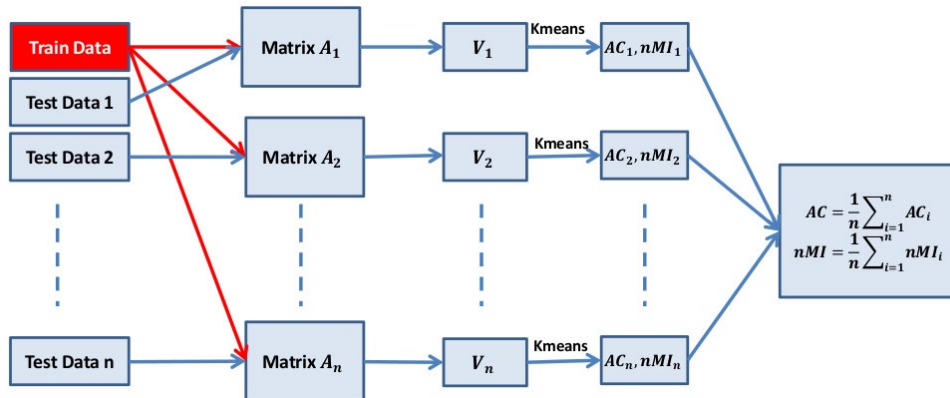


Figure 4.2: A schematic view of the proposed divide-and-conquer approach to get a new representation of the data for clustering. Here, the training data is mixed with each part of test data and is fed into VNMF/CMNMF to get new representation  $\mathbf{V}$ . The k-means algorithm is applied on each  $\mathbf{V}$  separately and the results are mixed as the final results of clustering.

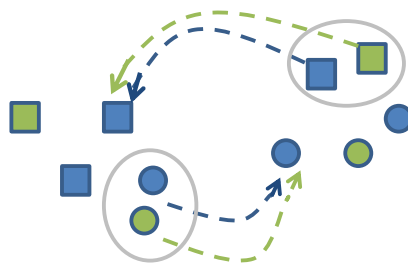


Figure 4.3: The objects in blue are the training data, and in green are test data. The square and circle indicates different classes. The dash line shows the similarity interaction. The blue dash interactions are done by the user while training the data. The green dash interactions are done by the system while applying the locality property

#### 4.3.4.4 Results

We compared the clustering results of new representations obtained from VNMF and CMNMF with that of the k-means clustering algorithm on original high-dimensional features, PCA, and NMF as a function of number of interactions and dimension of subspace separately. The three columns in both Figure 4.4 and Figure 4.5 show the results of SAR images represented by Mean Variance, Image Intensity, and WLD features, respectively.

Figure 4.4 shows the experimental results of clustering test data with the change of interaction number. As Figure 4.4 shows, by increasing the number of interactions, the clustering accuracy of VNMF and CMNMF for all data sets is increased by 10 – 15%. The user interaction provides more improvement in Mean-Variance and WLD features than Image Intensity features. For mutual information, all algorithms present similar performance within the range of  $\pm 3\%$ , which is reasonable for heuristic algorithms.

Figure 4.5 presents the experimental results of clustering test data with the change of dimension of subspace. Here, the dimension of subspace is the number of columns of new data representation  $\mathbf{V}$ . As shown in the Figure 4.5, with the increase of the dimension of new representations, all algorithms, except k-means, show the improvement in accuracy and mutual information. Among these algorithms, VNMF and CMNMF, shown in green and blue line respectively, offers better performance than other algorithms for all dimensions. It provides about 5 – 10% improvement over other algorithms. Compared among these three features, the user interaction improves the accuracy most for Mean-Variance features by more than 10%. For the mutual information, all algorithms have similar performances. Additionally, by observing the accuracy in Figure 4.5, we can find that for feature Mean-Variance and Image Intensity, once the dimension of subspace reaches 6, further increment in dimension of subspace cannot improve the clustering results in accuracy and mutual information. For WLD feature, after the dimension reaches 9, the accuracy and mutual information also reach their highest points in this dimension range. The results imply that, instead of setting the dimension of new representation to the number of classes, choosing some smaller values, like 6 for Mean-Variance and Image Intensity and 9 for WLD, will not affect the clustering result but will decrease the size of new representations substantially.

#### 4.3.4.5 Convergence

Figure 4.6 shows the convergence speed of NMF, VNMF and CMNMF for features Mean-Variance, Image Intensity and WLD.

As shown in figures above, these three algorithms converge within 10 iterations. VNMF provides a better objective value that is much smaller than that in NMF and CMNMF provides smallest objective value. Figure 4.6 shows the convergence speed

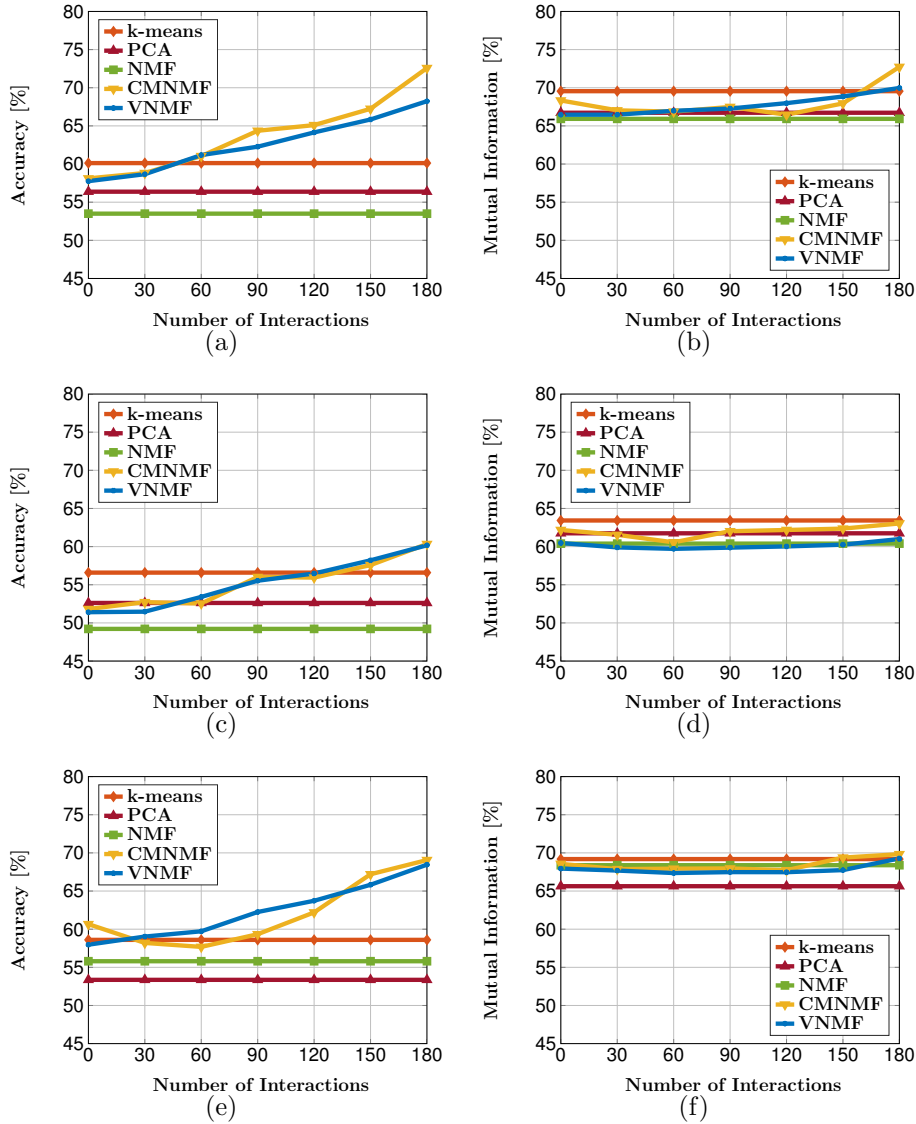


Figure 4.4: Clustering results of proposed algorithms VNMF and CMNMF as a function of number of interactions represented by accuracy (first column) and normalized mutual information (second column) compared with k-means clustering algorithm, PCA and NMF. The first, second, and third rows show the results of Mean Variance, Image Intensity and WLD features, respectively.

of Non-negative Matrix Factorization (NMF), VNMF, and CMNMF for features Mean-Variance, Image Intensity and WLD of SAR. As shown in figures above, these three algorithms converge within 10 iterations. VNMF provides a better objective value that is much smaller than that in NMF and CMNMF provides smallest objective value.

#### 4. Interactive Dimensionality Reduction

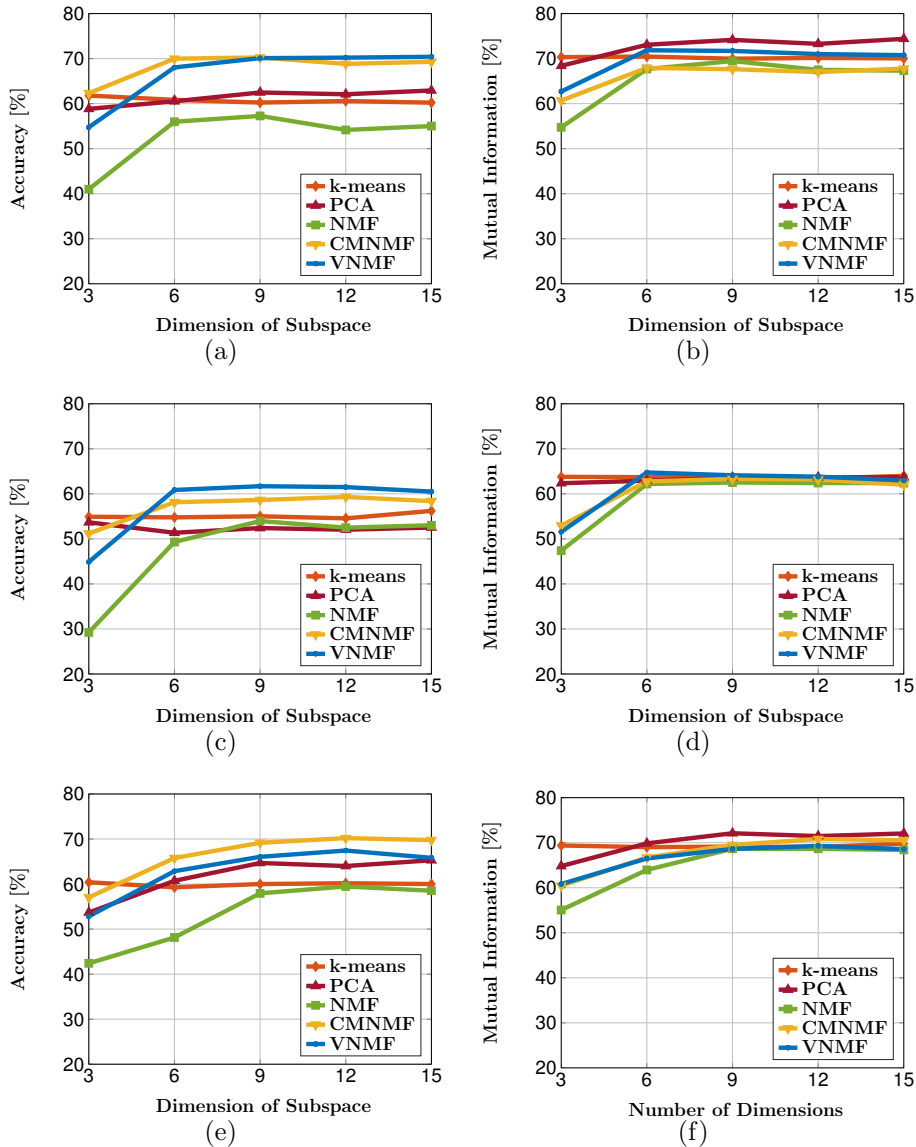


Figure 4.5: Clustering results of proposed algorithms VNMF and CMNMF as a function of dimension of subspace represented by accuracy (first column) and normalized mutual information (second column) compared with k-means clustering algorithm, PCA and NMF. The first, second, and third rows show the results of Mean Variance, Image Intensity and WLD features, respectively.

The computation times of divide-and-conquer and batch processing are depicted in Figure 4.7. The results confirm that the divide-and-conquer approach is about four times faster than batch processing. Fortunately, all matrix elements are updated independently from each other. Therefore, the factorization can be implemented on

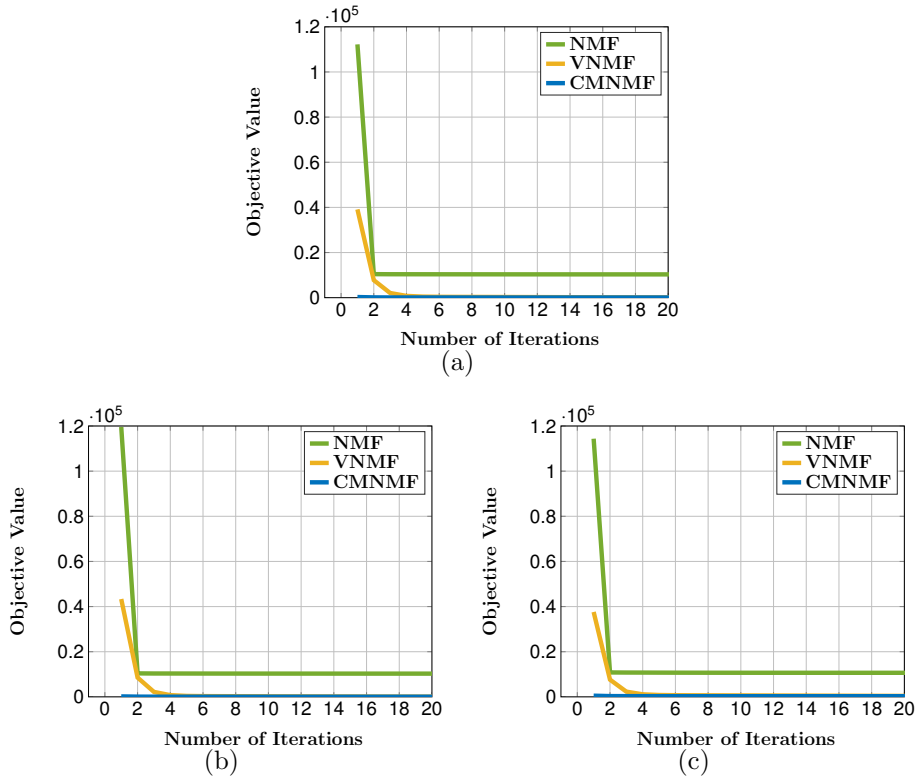


Figure 4.6: The convergence speed for NMF, VNMF and CMNMF applied to the SAR data set represented by a) Mean-Variance; b) Image Intensity; and c) WLD features.

GPU and consequently, the computation times can be decreased further.

## 4.4 Pair-wise constrained NMF

Having introduced two interactive algorithms, we propose another interactive algorithm for dimensionality reduction based on NMF. As we mentioned before, in content based image retrieval, the content of an image is represented by a feature vector, where these vectors have non negative values since they are built from the histogram of local features. Basically, the similarity between two images is measured by a kind of distance (e.g., Euclidean distance) between their corresponding feature vectors. However, semantically similar images might be interpreted as dissimilar images by the machine and vice versa (i.e., *semantic gap*). To bridge this semantic gap, we propose an NMF based algorithm with the following two semantic constraints:

1. Dissimilar images (in the view of machine) should be close together in the new representation, if they are semantically similar.

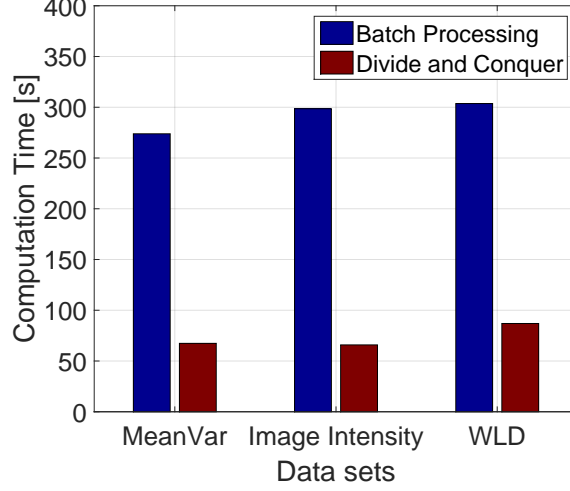


Figure 4.7: Computation times of divide-and-conquer compared to batch processing for all three datasets. The experiments were executed on a desktop PC with an Intel Core2Quad 2,8GHz CPU and 8GB of RAM. The divide-and-conquer approach is on average four times faster than batch processing.

2. Similar images (in the view of machine) should be far away in the new representation, if they are semantically dissimilar.

Here, we formulate the two aforementioned constraints, namely similarity preserving and fairness preserving, as regularization terms for the NMF. For both similarity and dissimilarity, we build two adjacency matrices  $\mathbf{W}$  and  $\mathbf{W}^{(f)}$ , respectively, initialized with 0. If image  $i$  is connected with image  $j$  as similar images, then  $\mathbf{W}_{ij} = 1$ . Conversely, if image  $i$  is connected with image  $j$  as dissimilar images, then  $\mathbf{W}_{ij}^{(f)} = 1$ . By coupling this constraint to the main function of NMF, we reach the following objective function:

$$O_1 = \frac{\lambda_1}{2} \sum_{i,j} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \mathbf{W}_{ij} = \lambda_1 \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}), \quad (4.23)$$

where  $\mathbf{L} = \mathbf{D} - \mathbf{W}$  and  $\mathbf{D}$  is a diagonal matrix whose entries are column sums of  $\mathbf{W}$ ,  $\mathbf{D}_{jj} = \sum_l \mathbf{W}_{jl}$ . Such a cost function has been used before for the purpose of structure preservation in [Cai+11]. For dissimilar images, we would like that their new representation be far away from each other. Therefore, we formulate another cost function:

$$\begin{aligned} O_2 &= \lambda_2 \exp \left[ -\frac{\beta}{2} \sum_{j,l} \|\mathbf{v}_j - \mathbf{v}_l\|^2 \mathbf{W}_{jl}^{(f)} \right] \\ &= \lambda_2 \exp \left[ -\beta \text{Tr}(\mathbf{V}^T \mathbf{L}^{(f)} \mathbf{V}) \right], \end{aligned} \quad (4.24)$$



where  $\mathbf{L}^{(f)} = \mathbf{D}^{(f)} - \mathbf{W}^{(f)}$  and  $\mathbf{D}^{(f)}$  is a diagonal matrix, whose entries are column sums of  $\mathbf{W}^{(f)}$ ,  $\mathbf{D}_{jj}^{(f)} = \sum_l \mathbf{W}_{jl}^{(f)}$ .

Adding the introduced terms to the NMF formulation leads to the following minimization objective:

$$O = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 + \lambda_1 \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \lambda_2 \exp[-\beta \text{Tr}(\mathbf{V}^T \mathbf{L}^{(f)} \mathbf{V})], \quad (4.25)$$

where  $\mathbf{X}$ , the original data, is decomposed to  $\mathbf{U}$  and  $\mathbf{V}$ , where  $\mathbf{V}$  is the new representation of the data. Using the KKT-conditions [BV09], we arrive at the following update rules for  $\mathbf{U}$  and  $\mathbf{V}$ :

$$u_{ik} \leftarrow u_{ik} \frac{(\mathbf{X}\mathbf{V})_{ik}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}} \quad (4.26)$$

$$v_{jk} \leftarrow v_{jk} \frac{(\mathbf{X}^T\mathbf{U})_{jk} + \lambda_1(\mathbf{L}^-\mathbf{V})_{jk} + \lambda_2\beta R^{(f)}(\mathbf{L}^{(f)+}\mathbf{V}^T)_{jk}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{jk} + \lambda_1(\mathbf{L}^+\mathbf{V})_{jk} + \lambda_2\beta R^{(f)}(\mathbf{L}^{(f)-}\mathbf{V}^T)_{jk}}, \quad (4.27)$$

where we introduced the terms  $\mathbf{L}^{(f)} = \mathbf{L}^{(f)+} - \mathbf{L}^{(f)-}$  with  $\mathbf{L}_{ij}^{(f)+} = (|\mathbf{L}_{ij}^{(f)}| + \mathbf{L}_{ij}^{(f)})/2$ ,  $\mathbf{L}_{ij}^{(f)-} = (|\mathbf{L}_{ij}^{(f)}| - \mathbf{L}_{ij}^{(f)})/2$ ,  $\mathbf{L} = \mathbf{L}^+ - \mathbf{L}^-$  with  $\mathbf{L}_{ij}^+ = (|\mathbf{L}_{ij}| + \mathbf{L}_{ij})/2$ ,  $\mathbf{L}_{ij}^- = (|\mathbf{L}_{ij}| - \mathbf{L}_{ij})/2$  and  $R^{(f)} = \exp[-\beta \text{Tr}(\mathbf{V}\mathbf{L}^{(f)}\mathbf{V}^T)]$ .

The proof of convergence is given in Appendix B.5.

## 4.4.1 Experiment 2

### 4.4.1.1 Data sets

The data sets used in our experiments were 1) the Caltech data set and 2) the Corel data set.

the **Caltech data set** contains 9144 images in 102 different groups. SIFT [Low04] feature vectors are extracted from these images, where each image is described by a 128-dimensional vector. For the experiment, we used the images from the 10 biggest groups.

the **Corel data set** contains 1500 images in 15 different groups, where each group contains 100 images. SIFT features [Low04] were extracted from these images, leading to 128 dimensional feature vectors.

### 4.4.1.2 Results and discussion

We compared the results of k-means clustering on the high-dimensional space with the results of k-means clustering on the new space generated by the proposed algorithm

with an increasing number of user interactions. For the simulation of the labeling process, we mapped the data to the 3D space using PCA, as is done in the CAVE. For each number of user interactions, we then annotated for the similarity term the given number of image pairs that have the same label and highest distance from each other in the obtained 3D space. We also annotated for the dissimilarity term the given number of images that have different labels, but are closest to each other in the obtained 3D space. Adding a pair of similar images corresponds to connecting them with a green line in the CAVE while adding a pair of dissimilar images corresponds to connecting them with a red line (see Figure 4.8). In order to get the average results, we repeated the experiments 10 times for each data set and chose a random subset of 500 images from each data set. The parameters were selected for each data set by performing cross-validation and selecting of the parameters that produce the best results. The new dimension of NMF,  $K$ , was set equal to the number of classes for each data set. Every time the k-means algorithm was applied, it was repeated 10 times and the best result was selected. The results are depicted in Figure 4.9. Clearly, the proposed algorithm outperforms the k-means algorithm for all data sets as the number of user interaction increases.

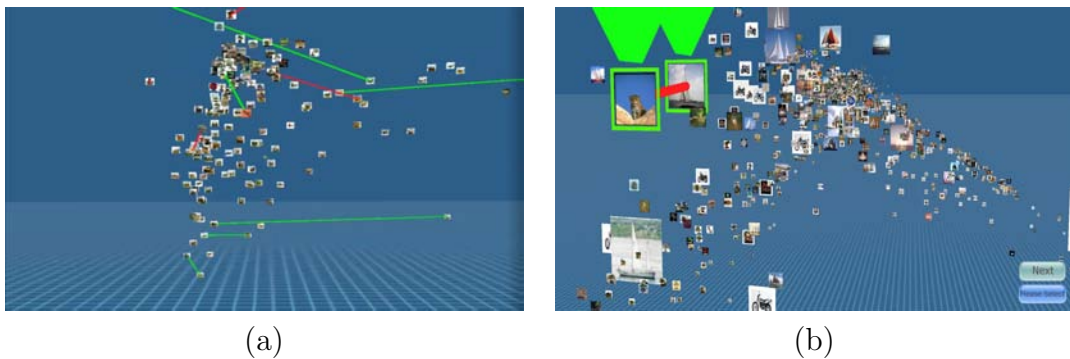


Figure 4.8: 3D visualization of the data sets using PCA and connecting similar and dissimilar images with green and red links. a) snapshot of interface from a far distance. b) snapshot of interface from a close distance. Here, two images that are close together but don't belong to the same class are linked with a red line.

## 4.5 Set-wise constrained NMF

In this method, once the images are visualized in the CAVE (by applying dimensionality reduction algorithm to the features), the user aims at creating several convex hulls around groups of similar images. The images that are inside one convex hull are assigned a single label. These labels are used to create a label matrix  $\mathbf{Q}$  to be used in the Discriminative Non-negative Matrix Factorization (DNMF) algorithm

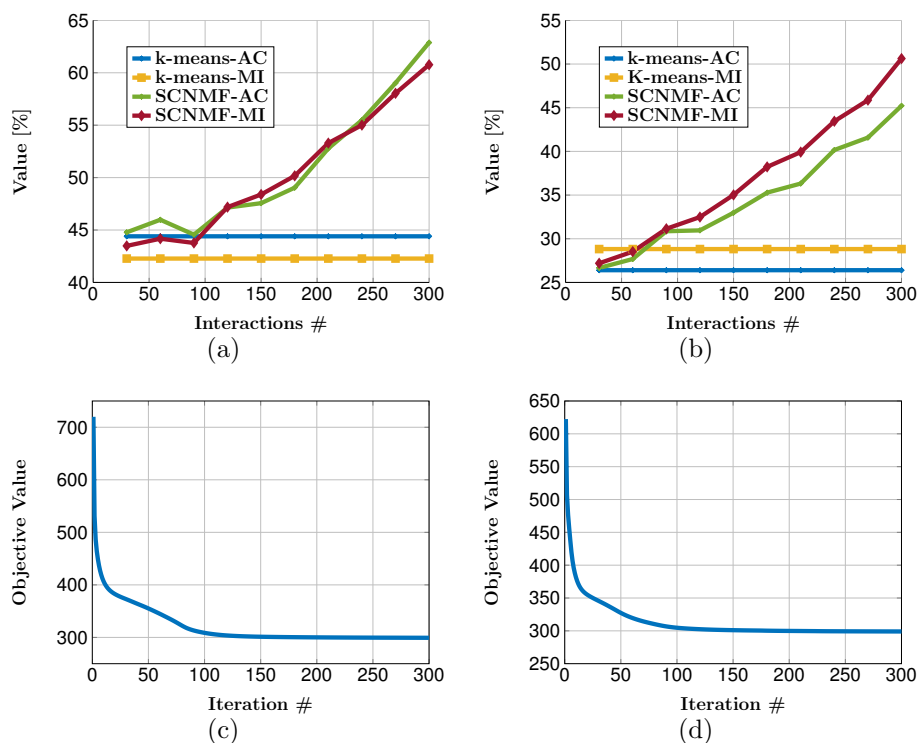


Figure 4.9: Clustering results for Caltech and Corel data sets. a) Caltech; b) Corel; c) The convergence rate of algorithm applied to the Caltech data set; d) The convergence of algorithm applied to the Corel data set.

that was explained in Chapter 2. Clearly, both the number of convex hulls and the amount of images in each convex hull depend on the type of features. Additionally, we also use PCA and NMF to create new representations. In the experiments, we choose at each time a subset of  $K$  clusters and set the dimension of PCA, NMF, and DNMF equal to  $K$ . The k-means algorithm is applied to do the clustering on these new representations. The clustering accuracy is used to demonstrate the performance of each representation. Figure 4.10 shows two snapshots of the user inside the CAVE while creating several convex hulls around similar images. In Table 4.1, the statistics of created convex hulls for the SAR data set is presented. As the Table shows, we have here 7 different sets that contain images from 6 different classes, namely  $C_1, C_2, \dots, C_6$ . The second column show the number of images in each set. For instance set 1 contains 16 images that 93.75% of them belong to class  $C_1$  and the rest belong to class  $C_2$ . This table also shows how our created convex sets contain the images from one specific class. If one convex set contains images from many different classes, then this would not be a good set. A good set is a set that contains images from one class. As an example, Convex set number 7 contains images from all different classes.

#### 4. Interactive Dimensionality Reduction

Set	Number of Images	C1	C2	C3	C4	C5	C6
1	16	93.75	6.25	0	0	0	0
2	59	1.69	23.73	74.58	0	0	0
3	45	2.2	95.56	2.22	0	0	0
4	266	6.39	4.89	2.26	67.29	1.13	18.05
5	292	14.38	.34	85.27	0	0	0
6	91	87.91	6.59	1.1	3.3	1.1	0
7	147	8.84	2.04	4.08	6.12	3.4	75.51

Table 4.1: The statistics of created convex hulls. The percentage of similar images (C1-C6) in each convex hull is presented. Additionally, the total number of images in each set is given in the second column. The seven created convex hulls contain different number of images. The images are coming from 6 different classes (C1-C6).

As the results show, by increasing the number of classes, the accuracy of clustering decreases. Both normalized Mutual Information and Accuracy of clustering are decreasing for all methods. However, our proposed algorithm outperforms the PCA and NMF algorithms. This improvement is more significant and visible for SIFT features. It should be noticed that the performance of this algorithm is weaker than the performance of CMNMF and VNMF. The main reason is that the created convex sets are enough accurate. The more accurate are the sets, the better performance we can anticipate from the algorithm. The advantage of this algorithm is that the user can select many images in a short time and assign label. This is useful in Active Learning scenarios, where the goal is to annotate the data point while learning a classifier.

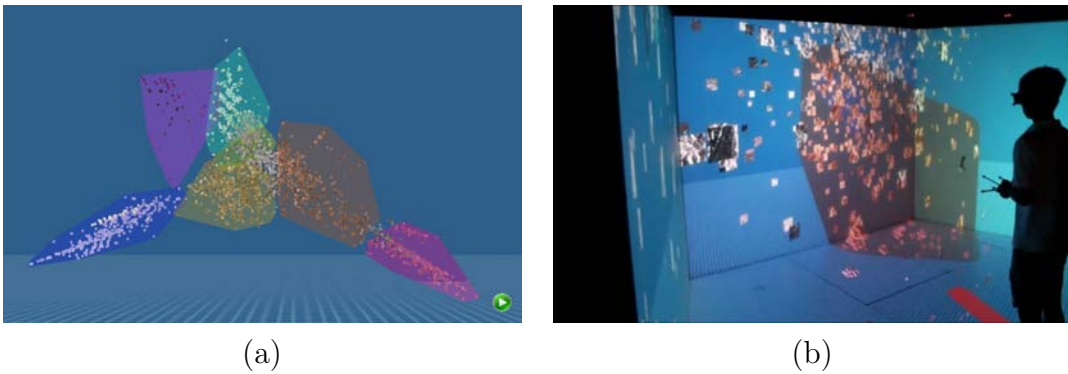


Figure 4.10: Sample snapshots of creating convex hulls (sets) around similar images in a Virtual Reality environment. a) a desktop display is used for visualization and interaction; b) the CAVE is used for creating sets.

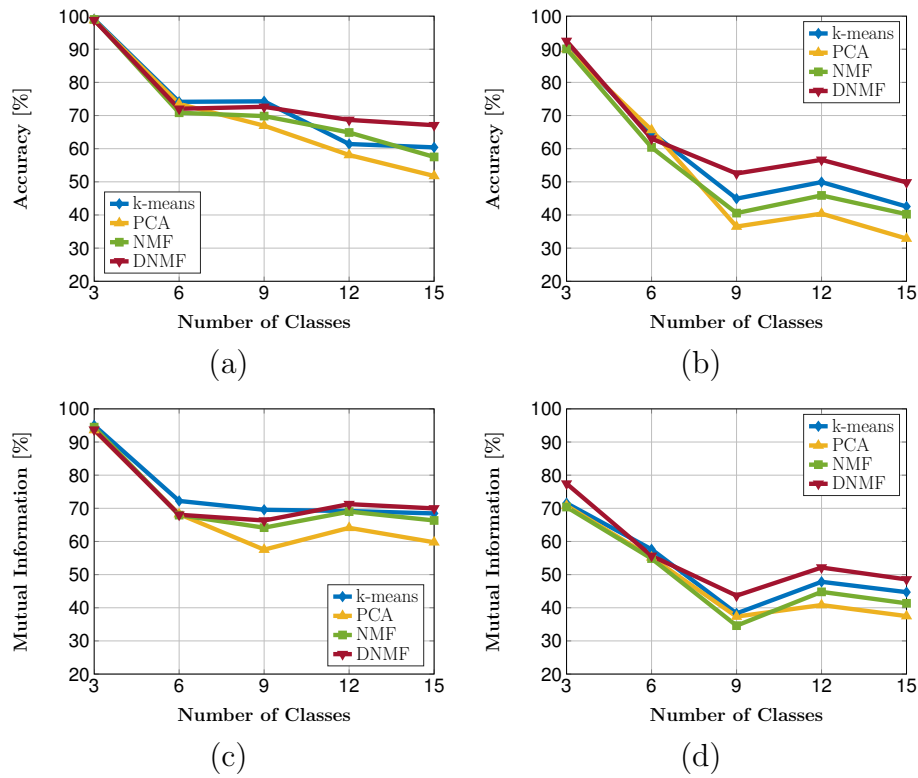


Figure 4.11: Clustering results for the SAR data set represented by WLD and SIFT features. First and second row show the accuracy and normalized mutual information, respectively. The first and second columns are the results of WLD and SIFT features, respectively.

## 4.6 Summary and conclusions

In this chapter, several interactive dimensionality reduction algorithms have been proposed. These techniques have two things in common: 1) all algorithms are devised in the context of non-negative matrix factorization by incorporating user interactions as constraints coupled to the main objective function of NMF and 2) an immersive 3D virtual reality is used as the user interface to support user-data interaction. Three types of interactions have been defined which are: 1) the images are visualized as a set of clusters obtained by k-means algorithm and the user links a mis-clustered image to the target cluster. CMNMF and VNMF are two interactive algorithms that use this interface; 2) the images are visualized using PCA and then pairs of similar and dissimilar images are linked, where Pair-wise NMF is used as DR algorithm; 3) the images are visualized using a dimensionality reduction technique and the user searches for a group of similar images and then draws a convex hull around them. A set-wise constrained NMF that utilizes the obtained convex sets is used to reduce the dimensionality. We performed several experiments for each technique and demonstrated that CMNMF and VNMF deliver better performance in comparison to the other algorithms. As future work, one can use the introduced constraints and user interface in the context of kernel learning to reduce the dimensionality of the data.

## Active Learning

Today we are dealing with a phenomenon, known as Big Data, where the amount of collected data has been increasing exponentially since the last decade [MC13]. Additionally, the complexity of the data is very high such that, usually, very high-dimensional features represent the data content. Automatic storage and retrieval of this data requires large-scale learning algorithms that rely on a large set of labeled data for training. On the one hand, providing labeled data is expensive and time consuming. On the other hand, unlabeled data is freely and cheaply available on a large scale. Therefore, active learning has gained much attention due to its ability to label the data and to train the classifier simultaneously [Set10; Tui+11; Set12; Per+14; Pat14].

In this chapter, we address the problem of active learning to annotate Synthetic Aperture Radar (SAR) image repositories. This chapter is mainly based on our recent article published in *IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing (JSTARS)* [Bab+15b]. The main contributions of this chapter are:

- to provide a review of active learning;
- to study the state-of-the-art active learning algorithms;
- to employ a trace-norm regularizer classifier as the training model in an active learning framework to annotate the SAR images;
- to study in depth the trace-norm classifier;
- to introduce a novel active learning algorithm based on visualization; and
- to conduct experiments to study the performance of the proposed active learning algorithm.

The rest of the chapter is organized as follows. Section 5.1 provides an overview of the concept of active learning. In Section 5.2, we review the state-of-the-art active learning algorithms. We illustrate the concept of multiclass classification with the trace-norm regularized classifier in Section 5.3.1. Here, we first study the proposed classifier and compare it in depth with Support Vector Machines (SVM). Then, we formulate one active learning algorithm solely based on this classifier. In Section 5.3.3, we introduce a novel active learning algorithm. Section 5.4 demonstrates our experiments conducted on a real SAR image data set. Finally, we provide a summary of the chapter in Section 5.5.

## 5.1 A review of active learning

The goal of active learning is to label a pool of unlabeled data and simultaneously train a classifier to categorize them. First, a set of labeled data  $\mathcal{L}$  is used to initially train a classifier (e.g., SVM). Next, a subset of unlabeled data  $\mathcal{U}$  along with their predicted label (computed by the classifier) is selected to query the true labels from an oracle (the user). The user examines the predicted labels and re-labels the data points (if necessary) and adds them to the pool of labeled data for retraining the classifier. This loop continues until all unlabeled data are labeled (annotated) and thus the classifier is trained. A schematic of the active learning cycle is provided in Figure 5.1. The three components impacting the performance of an active learning system are: 1) the choice of the training model; 2) the sample selection strategy; and 3) the user interface used for the communication of the user and the algorithm (i.e., interactive visualization). Successful active learning relies on proper combination of these components to handle huge volume of data. For instance, as the amount of data increases, the performance of the classifier does not improve significantly. More specifically, the challenge of informative sample selection is important when the amount of labeled data is low. Once the volume of labeled data increases, the over-fitting problem in the optimization process of the classifier appears.

There are three well-known active learning scenarios which are: 1) membership query synthesis; 2) stream-based selective sampling; and 3) pool-based active learning [Set10]. Figure 5.2 provides an overview of these scenarios along with existing sample selection strategies.

### 5.1.1 Membership query synthesis

Perhaps the first proposed active learning scenario is Membership Query Synthesis (MQS), where the learning program might query an instance from the input space, including synthesized samples that do not actually exist in the training data [Set10]. This scenario is mainly used to predict the absolute coordinates of a robot, given



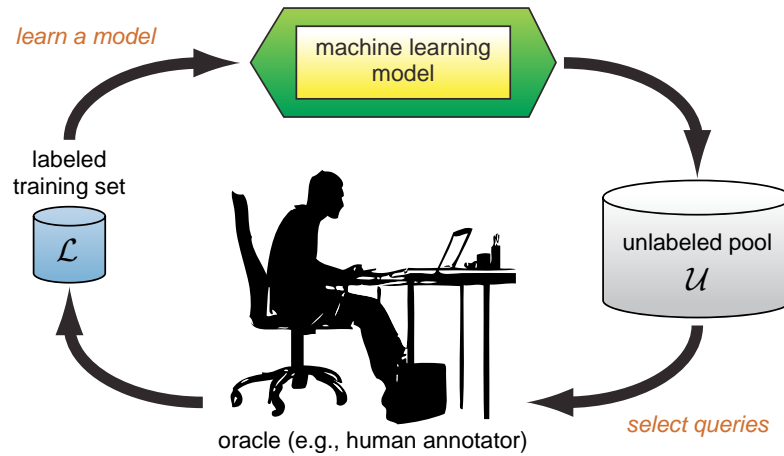


Figure 5.1: The active learning cycle (source [Set10])

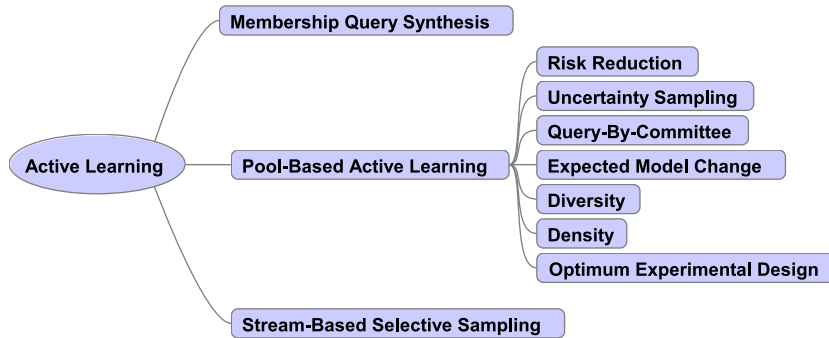


Figure 5.2: Overview of active learning scenarios and sample selection strategies

the joint angles of its mechanical arm or the autonomous execution of biological experiments. However, it also has been applied to classification and recognition problems, in which synthesized unrecognizable images or unreadable texts are given to the user to annotate [Set10].

### 5.1.2 Stream-based selective sampling

Stream-Based Selective Sampling (SBSS) is suitable for situations in which obtaining an unlabeled instance from the pool of unlabeled data set is inexpensive, while obtaining a labeled instance is involved with a high cost on the part of the human [Set10]. Therefore, the algorithm sequentially draws samples from the pool of unlabeled data and decides whether or not to label them. Since the samples are often drawn from the data set one at a time, this approach is also called stream-based selective sampling. The main challenge in stream-based selective sampling is how to make the decision. One solution is based on the prediction certainty of a trained classifier

for this sample. For instance, it can be done by defining a certainty threshold and querying all samples whose certainty is below a threshold. Moreover, it can be done based on training different classifiers on the same available labeled data and querying those unlabeled samples on which the classifiers disagree. Stream-based selection has been successfully applied to part-of-speech tagging, sensor scheduling, and word sense disambiguation problems.

### 5.1.3 Pool-based active learning

Pool-Based Active Learning (PBAL) assumes that a big pool of unlabeled data  $\mathcal{U}$  is available for free and all data samples are accessible at the same time [Set10]. This scenario exists in many real world applications such as image or document classification. Basically, the pool of unlabeled data  $\mathcal{U}$  does not change during the training process. The main challenge in PBAL is to select the most informative samples from the pool of unlabeled data and to add them to the pool of labeled data  $\mathcal{L}$ . One solution is to define a certainty measure and then to select the least certain sample, which imposes the most change into the classifier model. Another possible solution is to select a sample that has the best representation of the whole data distribution, e.g. based on clustering or statistical properties. The main difference between SBSS and PBAL is the amount of accessible samples from the pool of unlabeled data. Both PBAL and SBSS are often applicable to the same scenario and the decision of which of the two to use depends on the training circumstances, on memory restrictions, or on the availability of the whole data set at the same time. PBAL is preferred in text classification, information extraction, image classification, video classification, speech recognition, and cancer diagnosis [Set10].

Generally, two big groups of strategies exist for the selection of samples in pool-based active learning [WH11]. The first one is the selection of samples based on some properties of the current trained model. Strategies from this group are the following: 1) risk reduction selects samples that reduce the empirical risk of the classifier; 2) uncertainty sampling selects samples that the classifier is least certain about; 3) query by committee selects samples on which a committee of classifiers disagrees with most; and 4) expected model change selects samples that are expected to introduce the biggest change in the training model. The second group contains strategies that select samples based on their spatial distribution and ignore the current state of the training model. Strategies from this group are: 1) diversity-based selection, which involves selecting the most diverse samples; 2) density-based selection, which selects samples around which the data distribution is very dense; and 3) optimum experimental design, which selects samples that minimize the variance of a metric derived from the data set. The detailed description of these strategies are presented in the following sections.

### 5.1.3.1 Risk reduction

Expected risk minimization is a widely used criterion in machine learning [WH11], which is defined by:

$$\int_{\mathbf{x}} \mathbb{E}_{\mathbf{T}} [(\hat{y}(\mathbf{x}) - y(\mathbf{x}))^2 | \mathbf{x}] p(\mathbf{x}) d\mathbf{x} \quad (5.1)$$

where  $y(\mathbf{x})$  is the true label of  $\mathbf{x}$ ,  $\hat{y}(\mathbf{x})$  is the label predicted by the classifier,  $p(\mathbf{x})$  is the probability density function for  $\mathbf{x}$  and  $\mathbb{E}_{\mathbf{T}}$  denotes the expectation over the training data and possible label values. This leads to the idea of extending the goal of empirical risk minimization to active learning by computing the empirical risk reduction from the labeling of each sample and selecting the sample that reduces empirical risk the most. One method for achieving this was proposed in [CZJ96] by decomposing the risk into three terms:

$$\begin{aligned} \mathbb{E}_{\mathbf{T}} [(\hat{y}(\mathbf{x}) - y(\mathbf{x}))^2 | \mathbf{x}] = & \mathbb{E} [(y(\mathbf{x}) - \mathbb{E}[y(\mathbf{x})])^2 | \mathbf{x}] + \mathbb{E}_{\mathbf{L}} [(\hat{y}(\mathbf{x}) - \mathbb{E}[y(\mathbf{x})])^2 | \mathbf{x}] \\ & + \mathbb{E}_{\mathbf{L}} [(\hat{y}(\mathbf{x}) - \mathbb{E}_{\mathbf{L}}[\hat{y}(\mathbf{x})])^2 | \mathbf{x}], \end{aligned} \quad (5.2)$$

where  $\mathbb{E}_{\mathbf{L}}$  is the expectation over the labeled data and  $\mathbb{E}$  is the expectation over the conditional density  $P(y|\mathbf{x})$ . The first term describes the variance over the true label  $y$  based on  $\mathbf{x}$ . The second term describes the prediction error induced by the model and the third term describes the mean squared error of prediction with respect to the true model. The authors in [CZJ96] propose to estimate the reduced variance after adding each sample to the training set and to select the sample that minimizes the variance term the most. However, estimating the expected variance reduction is a challenging problem and very expensive for most classifiers, especially when we apply them to very large-scale data sets [WH11].

### 5.1.3.2 Uncertainty sampling

The uncertainty strategy chooses those samples for labeling that the classifier is least certain about. The certainty measure can be defined for most classifiers. For example, for binary classifiers, which estimate a boundary between different classes, the uncertainty can be defined based on the distance to the classifier boundary. Here, the classifier is most certain about samples that are far away from the boundary and least certain about samples that are close to the boundary. SVM<sub>Active</sub> [TC01], which is coupled to a SVM classifier, is the most well-known algorithm in this category. Moreover, for the most binary classifiers, a probability  $p(y_i|\mathbf{x})$  is defined to estimate the probability that sample  $\mathbf{x}$  has label  $y_i$ . Therefore, one typical uncertainty measure is the entropy [WH11]:

$$\text{certainty}(\mathbf{x}) = - \sum_i p(y_i|\mathbf{x}) \log p(y_i|\mathbf{x}). \quad (5.3)$$

and the sample selection strategy is selecting the samples with the highest entropy. For multiclass classification problems, one popular strategy is to select the samples based on the probabilities of the best and second best predictions [JPP09] defined by:

$$\text{certainty}(\mathbf{x}) = p(y_1|\mathbf{x}) - p(y_2|\mathbf{x}), \quad (5.4)$$

where  $y_1$  and  $y_2$  denote the classes with highest and second highest probability, respectively. If this difference is small, the model is less certain about which class is the correct one for that sample.

### 5.1.3.3 Query-by-committee

Query-By-Committee (QBC) is based on creating a committee of different training models  $\mathcal{C} = \{\theta^{(1)}, \dots, \theta^{(C)}\}$ , which are trained on the labeled data set  $\mathcal{L}$  [SOS92]. Therefore, each committee member has different predictions for input samples. Next, each committee member votes on each sample based on its predicted label and finally the sample which causes the most disagreement within the committee is chosen for labeling. The degree of disagreement between committee members presents an uncertainty degree for the sample label. Hence, QBC is also mentioned in the framework of uncertainty sampling. We should note that the key points are the choice of committee members, the committee size, and the disagreement measure. Basically, there are two approaches for measuring the disagreement degree. The first one is the vote entropy [Set10] defined by:

$$-\sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}, \quad (5.5)$$

where  $i$  ranges over all possible labels,  $V(y_i)$  denotes the number of committee members that voted for this label and  $C$  is the total number of committee members. The second approach is computing the average Kullback-Leibler Divergence (KLD) over all committee members [Set10] computed by:

$$\frac{1}{C} \sum_{i=1}^C D(P_{\theta^{(c)}}|P_C), \quad (5.6)$$

where

$$D(P_{\theta^{(c)}}|P_C) = \sum_i P(y_i|\mathbf{x}; \theta^{(c)}) \log \frac{P(y_i|\mathbf{x}; \theta^{(c)})}{P(y_i|\mathbf{x}; C)} \quad (5.7)$$

and  $\theta^{(c)}$  denotes a committee member. Thus, this measure picks in each iteration the sample for labeling whose average difference between the decision of a committee member and the whole committee is maximum.

### 5.1.3.4 Expected model change

Expected Model Change (EMC) aims at picking those samples that impose the biggest possible change onto the training model (i.e., classifier) [Set10]. Those classifiers that are based on gradient descent, and the EMC is directly predictable by the length of the training gradient once the sample is labeled. Since the true label of the sample is not known before labeling, the training gradient can be predicted by computing the expected value of its magnitude over all possible labels. Let  $l(\mathcal{L}; \theta)$  denote the optimization function of the classifier with parameter  $\theta$  with respect to the training set  $\mathcal{L}$  and  $\nabla l(\mathcal{L} \cup \langle \mathbf{x}, y_i \rangle; \theta)$  denote the resulting gradient after sample  $\mathbf{x}$  with label  $y_i$  is added to the training set. Then, the expected magnitude of the training gradient is given by [SCS08]:

$$\sum_i P(y_i | \mathbf{x}; \theta) \|\nabla l(\mathcal{L} \cup \langle \mathbf{x}, y_i \rangle; \theta)\|, \quad (5.8)$$

where  $P(y_i | \mathbf{x}; \theta)$  denotes the probability of sample  $\mathbf{x}$  having label  $y_i$ , based on the classifier with parameter  $\theta$ . The sample with the highest expected gradient magnitude is then added to the set of labeled samples. Although this approach has been shown to perform well in experiments, it can become very expensive if the dimension of feature vectors or the number of labels is large [Set10].

### 5.1.3.5 Diversity

The diversity criterion is especially important in cases where multiple samples are selected for labeling during each iteration. Reasons for this might be that the retraining of the model is expensive or that multiple humans are available for labeling. In these cases, in addition to previously introduced metrics such as uncertainty minimization, it is also desirable that the selected samples are as diverse as possible [WH11]. Given a distance metric  $K$ , the angle between two samples is defined as:

$$\cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{|K(\mathbf{x}_i, \mathbf{x}_j)|}{\sqrt{K(\mathbf{x}_i, \mathbf{x}_i)K(\mathbf{x}_j, \mathbf{x}_j)}} \quad (5.9)$$

Based on this, one way to estimate the diversity is [WH11]:

$$\text{Diversity}(\mathbf{x}) = 1 - \max_{\mathbf{x}_i} \frac{|K(\mathbf{x}_i, \mathbf{x})|}{\sqrt{K(\mathbf{x}_i, \mathbf{x}_i)K(\mathbf{x}, \mathbf{x})}}. \quad (5.10)$$

Another popular method for diversity estimation is based on Shannon entropy. For a set of points  $\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  the empirical entropy is defined as [DRH06]:

$$h(\mathcal{S}) = -\frac{1}{n} \sum_{i=1}^n \log \left( \frac{1}{n} \sum_{j=1}^n K(\mathbf{x}_i, \mathbf{x}_j) \right). \quad (5.11)$$

Then, those samples are selected that maximally reduce empirical entropy. The diversity criterion is usually not used by itself, but as a weighted combination of diversity and one of the previously introduced metrics.

### 5.1.3.6 Density

The density measure is based on the fact that points of high density regions in the feature space are usually more representative than the points of low density regions in a data set [WH11]. One way to estimate the density is to use a kernel density function [Zha+09a]. Given a Kernel function  $K(\mathbf{x})$  with the properties  $K(\mathbf{x}) > 0$  and  $\int K(\mathbf{x})d\mathbf{x} = 1$  and a set of points  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , a probability density function is first defined by

$$\hat{p}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n K(\mathbf{x} - \mathbf{x}_i). \quad (5.12)$$

Next, the density measure is defined by normalizing  $\hat{p}(\mathbf{x})$  to  $[0, 1]$  as follows [WH11]:

$$\text{Density}(\mathbf{x}) = \frac{\sum_{j=1}^n K(\mathbf{x} - \mathbf{x}_j)}{\max_i \sum_{j=1}^n K(\mathbf{x} - \mathbf{x}_j)}. \quad (5.13)$$

Finally, the points with the highest density are chosen for labeling. Similar to the diversity, the density measure is typically used in conjunction with one of the previously introduced metrics of uncertainty. Other density-based methods are those that first apply a clustering algorithm to the data set and then select those samples that are closest to the cluster centers [NS04; Qi+06]. Since the cluster centers are usually at the locations of highest density, these methods can also be considered density-based techniques [WH11].

### 5.1.3.7 Optimum experimental design

Optimum Experimental Design (OED) is an active learning method based on the statistical principle of variance minimization [ADT07]. In OED, each sample  $\mathbf{x}$  is called an experiment and its label  $y$  a measurement. An experimental design aims at designing experiments to minimize the variance of a parametrized model (here, classifier). There are two popular ways to employ experimental design in active learning: 1) select the samples minimizing the variance of the parameters of the training model and 2) select the samples minimizing the variance of the prediction value of the classifier. An experimental design is usually performed in combination with a linear regression model:

$$y = \mathbf{w}^T \mathbf{x} + \epsilon \quad (5.14)$$

with the parameter vector  $\mathbf{w} \in \mathbb{R}^d$  and the measurement noise  $\epsilon$  with zero mean and variance  $\sigma^2$ . Given the selected points matrix  $\mathbf{Z} = [\mathbf{x}_{s_1}, \dots, \mathbf{x}_{s_m}]^T$ , the optimal solution according to the least squares formulation is given by

$$\mathbf{w}^* = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y}, \quad (5.15)$$

and its covariance matrix by [ADT07]:

$$\text{Cov}(\mathbf{w}^*) = \sigma^2 (\mathbf{Z}^T \mathbf{Z})^{-1}. \quad (5.16)$$

One way to apply experimental design is to select samples that minimize some parameters of the covariance matrix of  $\mathbf{w}^*$ . For example, D-optimal design minimizes the determinant of  $\text{Cov}(\mathbf{w}^*)$ , A-optimality (average) design minimizes the trace of  $\text{Cov}(\mathbf{w}^*)$  and E-optimality (eigenvalue) design maximizes the minimum eigenvalue of  $\text{Cov}(\mathbf{w}^*)$  [ADT07]. Another way is to select samples that minimize the variance of the predicted value of the classifier (for a test point  $\mathbf{v}$ ), which is given by  $\mathbf{w}^{*T} \mathbf{v}$  and whose predictive variance is computed by

$$\mathbf{v}^T \text{Cov}(\mathbf{w}^*) \mathbf{v}. \quad (5.17)$$

Two common ways to minimize the above criteria are using I-optimal design to minimize the average predictive variance over a set of test points and using G-optimal design to maximize the predictive variance over a set of test samples [ADT07].

## 5.1.4 Training models

As we discussed earlier, most sample selection strategies select samples based on some properties of an associated training model (classifier). Hence, the classifier selection is a crucial part for a successful development of an active learning algorithm. In this section, we provide an overview of the most widely used classifiers in active learning. These classifiers are the following: 1) Support Vector Machine (SVM), which aims at estimating a classification boundary between different classes by determining support vectors; 2) Regularized Least Squares (RLS), which aims at estimating the classification boundary by minimizing the squared loss of a linear classifier over all samples; and 3)  $k$  Nearest Neighbors (k-NN), which predicts the label of a sample based on the labels of its  $k$  nearest neighbors.

### 5.1.4.1 Support vector machine

We first consider the maximum margin classifier for a binary classification problem. Let a set of samples  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ ,  $\mathbf{x}_i \in \mathbb{R}^d$  be assigned with the labels  $y_i \in \{-1, 1\}$  and a linear classifier  $\mathbf{w}^T \mathbf{x}$ . We initially assume that the samples from the two classes are linearly separable. Then, the labels predicted by the classifier are given by  $\text{sign}(\mathbf{w}^T \mathbf{x})$ . However, there are infinite by many options for selecting  $\mathbf{w}$  such that the

sign of all training labels can be predicted correctly. The maximum margin classifier suggests selecting a value for  $\mathbf{w}$  that maximizes the classification margin [CS00]. The margin is defined as the distance between the classification boundary and the closest sample that can be computed by constructing two hyperplanes passing through the closest samples from each class and parallel to the classification boundary. The two hyperplanes are defined by

$$\mathbf{w}^T \mathbf{x} = 1 \text{ and } \mathbf{w}^T \mathbf{x} = -1. \quad (5.18)$$

All training samples must lie outside the classifier margin, which means the following conditions should be fulfilled:

$$\begin{cases} \mathbf{w}^T \mathbf{x}_i \geq 1 & \text{if } y_i = 1, \\ \mathbf{w}^T \mathbf{x}_i \leq -1 & \text{if } y_i = -1. \end{cases} \quad (5.19)$$

Holding the above conditions is equal to holding the following condition

$$y_i(\mathbf{w}^T \mathbf{x}_i) \geq 1. \quad (5.20)$$

Therefore, the size of the margin is given by half the distance between the two hyperplanes that can be determined by geometric considerations to  $1/\|\mathbf{w}\|$ . Thus, maximizing the margin is equivalent to minimizing  $\|\mathbf{w}\|$  or the simpler form  $\frac{1}{2}\|\mathbf{w}\|^2$ . This leads to the following optimization problem for the maximum margin classifier:

$$\begin{aligned} &\text{minimize} && \frac{1}{2} \|\mathbf{w}\|^2 && (5.21) \\ &\text{subject to} && y_i(\mathbf{w}^T \mathbf{x}_i) \geq 1. \end{aligned}$$

The disadvantage of maximum margin classifier is its sensitivity to noise, mislabeled points or outliers. It might even be impossible to define a hyperplane that separates the data into two classes. To remedy this situation, a soft margin has been proposed that allows some of the training samples to slip into the margin to a certain degree [CS00]. In order to define a soft margin, non-negative slack variables  $\xi_i$  are introduced for each sample, and the condition (5.20) is replaced by

$$y_i(\mathbf{w}^T \mathbf{x}_i) + \xi_i \geq 1. \quad (5.22)$$

In order to penalize the effect of samples slipping into the margin, the optimization objective is extended by the weighted sum of all slack variables. This leads to the SVM optimization problem [CS00]:

$$\begin{aligned} &\text{minimize} && \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i && (5.23) \\ &\text{subject to} && y_i(\mathbf{w}^T \mathbf{x}_i) + \xi_i \geq 1 \\ &&& \xi_i \geq 0. \end{aligned}$$



The optimization problem (5.23) is a quadratic optimization problem. It can be either solved directly or in its dual form. The dual form is obtained by introducing non-negative Lagrange multipliers  $\alpha_i$  for the constraints (5.22) and applying the Karush-Kuhn-Tucker (KKT) conditions [BV09]. These lead to the following relationship between  $\mathbf{w}$  and the Lagrange multipliers:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i. \quad (5.24)$$

The dual optimization problem can then be derived to

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ & \text{subject to} && 0 \leq \alpha_i \leq C. \end{aligned} \quad (5.25)$$

The Lagrange multipliers determine the degree of which each sample contributes to the computation of  $\mathbf{w}$ . In practice, only some Lagrange multipliers are greater than zero. The samples corresponding to these Lagrange multipliers are called Support Vectors, which leads us to name the classifier SVM. So far, the type of classification boundary has been restricted to linear. It is also possible to apply the SVM to a nonlinear classification problem by using the kernel trick [CS00]. The kernel trick involves projecting the samples that are not separable by a linear boundary in the original space, to a space of higher dimension, where a linear boundary can separate them. This is achieved by applying a mapping function  $\phi(\mathbf{x})$  that projects each sample to the new space. In the optimization problem of SVM, the samples  $\mathbf{x}$  are then replaced by their mapped features  $\phi(\mathbf{x})$  in equations (5.23) or (5.25) and labels are predicted based on the product  $\mathbf{w}^T \phi(\mathbf{x})$ . In equation (5.25) the training samples are introduced into the optimization problem by the dot product  $\mathbf{x}_i \cdot \mathbf{x}_j$ . By applying the feature mapping, this is replaced by the dot product  $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ . It can be shown [CS00] that this is equivalent to applying a kernel function  $k(\mathbf{x}_i, \mathbf{x}_j)$  to the original space. The kernel function allows to implicitly introduce the high dimensional mapping, without actually performing the projection for the computations. This even makes it possible to use implicit mappings to infinite dimensional spaces. Some popular kernel functions are the polynomial kernel of degree  $p$ :

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^p, \quad (5.26)$$

the Gaussian Radial Basis Function (RBF)

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \quad (5.27)$$

with the parameter  $\gamma$  and the hyperbolic tangent:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j + c), \quad (5.28)$$

with the parameters  $\kappa > 0$  and  $c < 0$ .

Introducing the kernel function into the optimization problem (5.25) leads to the following new optimization problem:

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) && (5.29) \\ & \text{subject to} && 0 \leq \alpha_i \leq C. \end{aligned}$$

Additionally, the product  $\mathbf{w}^T \phi(\mathbf{x})$  can now be expressed by introducing the feature mapping into equation (5.24) as:

$$\mathbf{w}^T \phi(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j). \quad (5.30)$$

Another important extension of the SVM is its application to multiclass scenario. One popular strategy is the One-Versus-Rest (OVR) approach. For  $k$  different classes, it involves solving  $k$  different binary problems. For each class  $l$ , the samples of that class are labeled with 1 and the samples of all other classes with  $-1$ . Next, after training, a vector  $\mathbf{w}_l$  corresponding to the class  $l$  is obtained. The label of a sample  $\mathbf{x}$  is then predicted by

$$y = \arg \max_{l=1, \dots, k} \mathbf{w}_l^T \mathbf{x}. \quad (5.31)$$

#### 5.1.4.2 Regularized least squares

In Regularized Least Squares (RLS), the multiclass classification problem with  $k$  classes is treated as a regression problem [EPP00]. Let  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$  denote the training samples matrix and  $\mathbf{Y} \in \mathbb{R}^{n \times k}$  denote the label information matrix. Then, the matrix  $\mathbf{Y} = [Y_{i,j}]$  is defined as follows:

$$Y_{i,j} = \begin{cases} 1 & \text{if } y_i = j \\ 0 & \text{otherwise.} \end{cases} \quad (5.32)$$

The RLS involves training a classifier  $\mathbf{W} \in \mathbb{R}^{d \times k}$  by minimizing the sum of a regularization term and the squared difference of the predicted labels to the provided training labels [EPP00]:

$$\|\mathbf{Y} - \mathbf{XW}\|^2 + \gamma \|\mathbf{W}\|^2, \quad (5.33)$$

with the positive regularization parameter  $\gamma$ . This problem can be solved analytically by [EPP00]:

$$\mathbf{W} = (\mathbf{X}^T \mathbf{X} + \gamma \mathbf{I})^{-1} \mathbf{X}^T \mathbf{Y}. \quad (5.34)$$

The labels are then predicted for each sample by picking the column with the maximal value of the product  $\mathbf{x}^T \mathbf{W}$ :

$$y = \arg \max_{l=1, \dots, k} (\mathbf{x}^T \mathbf{W})_l. \quad (5.35)$$

As with the SVM, the RLS classifier can also be used in combination with a kernel function. The authors in [EPP00] claim that in this case the product  $\mathbf{x}^T \mathbf{W}$  becomes a linear combination of the kernel functions over all points:

$$(\mathbf{x}^T \mathbf{W})_l = \sum_{i=1}^n C_{i,l} k(\mathbf{x}_i, \mathbf{x}), \quad (5.36)$$

with the coefficient matrix  $\mathbf{C} = [C_{i,j}] \in \mathbb{R}^{n \times k}$ . This is obtained by minimizing the following function [EPP00]:

$$\|\mathbf{Y} - \mathbf{K}\mathbf{C}\|^2 + \gamma \text{Tr}[\mathbf{C}^T \mathbf{K}\mathbf{C}], \quad (5.37)$$

with the kernel matrix over all training samples  $\mathbf{K} = [K_{i,j}] \in \mathbb{R}^{n \times n}$  defined as

$$K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j). \quad (5.38)$$

The solution can be again computed analytically by

$$\mathbf{C} = (\mathbf{K} + \gamma \mathbf{I})^{-1} \mathbf{Y}. \quad (5.39)$$

### 5.1.4.3 k-nearest neighbors

The  $k$ -Nearest Neighbors (kNN) algorithm computes a sample label based on the labels of its  $k$  nearest neighbors [Alt92]. Given a test sample  $\mathbf{v}$ , a distance metric  $d(\mathbf{x}_1, \mathbf{x}_2)$  (e.g., euclidean distance), and the parameter  $k$ , the kNN computes the distances of all samples in the training set to the test sample

$$d(\mathbf{v}, \mathbf{x}_i), \quad (5.40)$$

and estimates the set  $\mathcal{N}_k = \mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k}$  of the  $k$  closest samples. The label of the test sample is determined based on the labels of its neighbors. Normally, the number of times each class occurs among the neighbors is counted, and next, the class that occurs most is picked as the label of the test sample. There is a disadvantage to this method. When the samples are not equally distributed among all classes, the dominant class is very often chosen for labeling since it has many occurrences among the neighbors. To resolve this issue, another way is to make the vote of each neighbor dependent on its distance to the test sample. Then, the weight for class  $i$  is given by [Alt92]

$$b_i = \sum_{k; y_{j_k} = i} \frac{1}{d(\mathbf{v}, \mathbf{x}_{j_k})} \quad (5.41)$$

and the label of  $\mathbf{v}$  by

$$\arg \max_i b_i. \quad (5.42)$$

Although the kNN classifier is fast and requires no training model, it is very sensitive to the local structure of the data and outliers.

## 5.2 State-of-the-art algorithms

Having provided an overview of different active learning scenarios and more specifically of general strategies in pool-based active learning and the associated classifiers, here we present several state-of-the-art active learning algorithms in more detail. These algorithms are: 1) Transductive Experimental Design (TED); 2) Locally Linear Reconstruction (LLR); 3) Manifold Adaptive Experimental Design (MAED); and 4)  $\text{SVM}_{\text{Active}}$ . Out of these methods, TED, LLR and MAED belong to the category of optimum experimental design and therefore select samples independently of the classifier. However, unless stated otherwise, it is assumed that the SVM classifier is used for training on the selected samples.  $\text{SVM}_{\text{Active}}$  belongs to the strategy of uncertainty-based active learning and as the name suggests, selects samples based on the uncertainty of the SVM classifier. We also used these algorithms as benchmark in our experiments and compared them with our proposed algorithms introduced in the next sections.

### 5.2.1 Transductive experimental design

The Transductive Experimental Design algorithm [YBT06] is a variant of experimental design algorithms [ADT07]. The key idea is to select points based on the minimization of a statistical variance metric of the trained classifier:

$$\mathbf{y} = \mathbf{w}^T \mathbf{x} + \epsilon. \quad (5.43)$$

TED considers the Regularized Least Squares formulation:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left\{ J = \sum_{i=1}^n (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \gamma \|\mathbf{w}\|^2 \right\}, \quad (5.44)$$

with the regularization parameter  $\gamma$ . The solution of this problem can be obtained by

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X} + \gamma \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}, \quad (5.45)$$

where  $\mathbf{I}$  is an identity matrix. Next, the variance of the predictive error can be estimated by [YBT06]:

$$\sigma^2 = \text{Tr}(\mathbf{X}(\mathbf{X}^T\mathbf{X} + \gamma\mathbf{I})^{-1}\mathbf{X}^T). \quad (5.46)$$

The goal is to select points  $\mathbf{Z} \subset \mathbf{X}$  that minimize the predictive variance. Therefore, the TED active learning problem is formulated as:

$$\begin{aligned} \min_{\mathbf{Z}} & \text{Tr}(\mathbf{X}(\mathbf{Z}^T\mathbf{Z} + \gamma\mathbf{I})^{-1}\mathbf{X}^T) \\ & \text{subject to } \mathbf{Z} \subset \mathbf{X}, |\mathbf{Z}| = m \end{aligned} \quad (5.47)$$

After some mathematical derivations [YBT06], the above problem can be transformed into

$$\begin{aligned} \min_{\mathbf{a}_i, \mathbf{Z}} & \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{Z}^T \mathbf{a}_i\|^2 + \gamma \|\mathbf{a}_i\|^2 \\ & \text{subject to } \mathbf{Z} \subset \mathbf{X}, |\mathbf{Z}| = m \end{aligned} \quad (5.48)$$

with  $\mathbf{a}_i \in \mathbb{R}^m$ ,  $i = 1, \dots, n$ . This formulation shows that TED tends to select the points that can linearly reconstruct the whole data set more precisely with the smallest possible coefficients  $\mathbf{a}_i$ . However, the problem (5.47) is NP-hard and thus is very difficult to solve [YBT06]. A sequential algorithm for finding an approximate suboptimal solution was proposed in [YBT06]. The authors in [Yu+08] propose a non-greedy algorithm, which solves the following relaxation problem (5.47):

$$\begin{aligned} \min_{\beta, \alpha_i \in \mathbb{R}^n} & \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{X}^T \alpha_i\|^2 + \sum_{j=1}^m \frac{\alpha_{i,j}^2}{\beta_j} + \gamma \|\beta\|_1 \\ & \text{subject to } \mathbf{x}_i \in \mathbf{X}, \beta_j \geq 0, j = 1, \dots, m \end{aligned} \quad (5.49)$$

where  $\alpha_i = [\alpha_{i,1}, \dots, \alpha_{i,n}]^T$ . This problem is convex and has a global minimum [Yu+08]. For its solution, the following update rules are derived, which are repeated until convergence:

$$\beta_j \leftarrow \sqrt{\frac{1}{\gamma} \sum_{i=1}^n \alpha_{i,j}^2} \text{ for } j = 1, \dots, m \quad (5.50)$$

$$\alpha_i \leftarrow (\text{diag}(\beta)^{-1} + \mathbf{X}\mathbf{X}^T)^{-1} \mathbf{X}\mathbf{x}_i \text{ for } i = 1, \dots, n. \quad (5.51)$$

There,  $\text{diag}(\beta)$  denotes the diagonal matrix whose diagonal elements are set to the corresponding elements of  $\beta$ .

## 5.2.2 Locally linear reconstruction

For those data sets that are coming from a manifold embedded in an ambient space, LLR has been proposed as an effective active learning algorithm [Zha+11]. The authors in [Zha+11] propose an algorithm that gains information about the manifold by computing a local reconstruction matrix and then selects those points that can most precisely reconstruct the whole data set based on the local reconstruction matrix. The reconstruction matrix was first introduced in [RS00] as part of the Locally Linear Embedding (LLE) algorithm and is computed by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{W} \in \mathbb{R}^{n \times n}} \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{j=1}^n w_{ij} \mathbf{x}_j \right\|^2 \\ \text{subject to } \sum_{j=1}^n w_{ij} = 1, \quad i = 1, \dots, n \\ w_{ij} = 0 \text{ if } \mathbf{x}_j \notin N_p(\mathbf{x}_i), \end{aligned} \quad (5.52)$$

where  $N_p(\mathbf{x})$  denotes the  $p$  nearest neighbors of  $\mathbf{x}$ . After the matrix  $\mathbf{W}$  has been computed, [Zha+11] introduces the inverse problem of reconstructing a set of points  $\{\mathbf{q}_1, \dots, \mathbf{q}_n\}$  based on the reconstruction matrix and a set of selected data points  $\{\mathbf{x}_{s_1}, \dots, \mathbf{x}_{s_m}\}$ . This is achieved by solving the following problem:

$$\min_{\mathbf{q}_1, \dots, \mathbf{q}_n} \sum_{i=1}^m \|\mathbf{q}_{s_i} - \mathbf{x}_{s_i}\|^2 + \mu \sum_{i=1}^n \left\| \mathbf{q}_i - \sum_{j=1}^n w_{ij} \mathbf{q}_j \right\|^2, \quad (5.53)$$

where  $\mu$  is a constant. In the cost function of this problem, the first term aims at fixing the selected points, while the second term aims at reconstructing the points based on the reconstruction matrix  $\mathbf{W}$ . By setting  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T$ ,  $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_n]^T$  and introducing the diagonal matrix  $\mathbf{\Lambda}$  with  $\Lambda_{ii} = 1$  if  $i \in \{s_1, \dots, s_m\}$ , this problem is transformed into the following form [Zha+11]:

$$\min_{\mathbf{Q}} \text{Tr}[(\mathbf{Q} - \mathbf{X})^T \mathbf{\Lambda} (\mathbf{Q} - \mathbf{X})] + \mu \text{Tr}(\mathbf{Q}^T \mathbf{M} \mathbf{Q}), \quad (5.54)$$

with  $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$  and its solution becomes

$$\mathbf{Q} = (\mu \mathbf{M} + \mathbf{\Lambda})^{-1} \mathbf{\Lambda} \mathbf{X}. \quad (5.55)$$

The authors in [Zha+11] propose to measure the representativeness of the selected points by the reconstruction error. This error is the difference between points  $\mathbf{Q}$  and the original points  $\mathbf{X}$ :

$$\begin{aligned} e &= \|\mathbf{X} - \mathbf{Q}\|_F^2 \\ &= \|(\mu \mathbf{M} + \mathbf{\Lambda})^{-1} \mu \mathbf{M} \mathbf{X}\|_F^2, \end{aligned} \quad (5.56)$$

where  $\|\cdot\|_F^2$  is the matrix Frobenius norm. This leads to the following Active Learning (AL) problem of selecting the most representative points:

$$\begin{aligned} \min \quad & \|(\mu\mathbf{M} + \mathbf{\Lambda})^{-1}\mu\mathbf{M}\mathbf{X}\|_F^2, \\ \text{s.t.} \quad & \mathbf{\Lambda} \in \mathbb{R}^{n \times n} \text{ is diagonal} \\ & \Lambda_{ii} \in \{0, 1\}, i = 1, \dots, n \\ & \sum_{i=1}^n \Lambda_{ii} = m, \end{aligned} \tag{5.57}$$

where, based on the matrix  $\mathbf{\Lambda}$ , the points with  $\Lambda_{ii} = 1$  are selected. However, due to its combinatorial nature, this problem is NP-hard and very difficult to solve. Therefore, [Zha+11] proposes two optimization schemes to compute an approximate solution of (5.57). The first one is a greedy sequential algorithm, which selects one sample at a time and the second is based on solving the convex relaxation of (5.57). Since the convex relaxation method has been shown to be very complex and time-consuming for large-scale data sets [Zha+11], only the sequential algorithm is described in this section.

In the sequential algorithm, we start with a set of  $l$  points  $\mathcal{Z} = \{\mathbf{x}_{s_1}, \dots, \mathbf{x}_{s_l}\}$  and the matrix  $\mathbf{\Lambda}_l$  as a diagonal  $n \times n$  matrix whose diagonal entries corresponding to the selected points  $\{s_1, \dots, s_l\}$  are set to 1 and the others to 0. Additionally, the matrix  $\mathbf{\Gamma}_i$  is defined as the  $n \times n$  matrix, whose diagonal entry at  $(i, i)$  is set to 1 and all other entries to 0. Next, the successive points to be added are found by solving the following problem:

$$s_{l+1} = \underset{i \notin \{s_1, \dots, s_l\}}{\operatorname{argmin}} \|(\mu\mathbf{M} + \mathbf{\Lambda}_l + \mathbf{\Gamma}_i)^{-1}\mu\mathbf{M}\mathbf{X}\|_F^2. \tag{5.58}$$

This process is repeated until the desired number of points are selected. The details of this algorithm are presented in [Zha+11].

### 5.2.3 Manifold adaptive experimental design

The MAED algorithm [CH12] is an active learning algorithm for data sets that lie in a manifold embedded in the high dimensional space. It is an extension of the TED algorithm [YBT06] with a manifold adaptive kernel [VP05], which allows the representativeness of the data points to be computed based on the geodesic distance on the manifold. The authors in [VP05] propose to incorporate manifold information into a classifier in order to achieve a better estimation of the classification boundary. Let  $\mathcal{H}$  represent the space of functions with the kernel  $\mathcal{K}$ . The manifold adaptive kernel space is the space  $\tilde{\mathcal{H}}$  with the kernel

$$\tilde{K}(\mathbf{x}, \mathbf{z}) = K(\mathbf{x}, \mathbf{z}) - \lambda \mathbf{k}_x^T (\mathbf{I} + \mathbf{M}\mathbf{K})^{-1} \mathbf{M}\mathbf{k}_z, \tag{5.59}$$

where  $\mathbf{M}$  is a positive definite matrix,  $\mathbf{I}$  is an identity matrix,  $\mathbf{K}$  is the kernel matrix in  $\mathcal{H}$ ,  $\lambda \geq 0$  is a weighting constant, and the vector  $\mathbf{k}_x$  is defined by:

$$\mathbf{k}_x = (\mathcal{K}(\mathbf{x}, \mathbf{x}_1), \dots, \mathcal{K}(\mathbf{x}, \mathbf{x}_n)). \quad (5.60)$$

The authors in [VP05] show that the space  $\tilde{\mathcal{H}}$  is still a Reproducing Kernel Hilbert Space (RKHS), where the matrix  $\mathbf{M}$  controls how the kernel space is deformed based on the intrinsic data geometry.

Cai et al. [CH12] suggest to set  $\mathbf{M}$  to the graph Laplacian. In order to compute the graph Laplacian, a nearest neighbor graph  $G$  is constructed first by drawing an edge between each data point  $\mathbf{x}_i$  and its  $k$  nearest neighbors  $\mathbf{x}_j$ . Next, a weight matrix  $\mathbf{W} \in \mathbb{R}^n \times \mathbb{R}^n$  is computed as:

$$w_{i,j} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are connected by an edge} \\ 0, & \text{otherwise.} \end{cases} \quad (5.61)$$

Finally, the graph Laplacian is defined as  $\mathbf{L} = \mathbf{D} - \mathbf{W}$ , with the diagonal matrix  $\mathbf{D}$  given by  $D_{ii} = \sum_j w_{ij}$ . Setting  $\mathbf{M} = \mathbf{L}$  leads to the following manifold adaptive kernel matrix  $\mathbf{K}_M = [K_{M;i,j}]$ :

$$K_{M;i,j} = K_{i,j} - \lambda \mathbf{k}_i^T (\mathbf{I} + \mathbf{L}\mathbf{K})^{-1} \mathbf{L}\mathbf{k}_j, \quad (5.62)$$

where  $\mathbf{K}$  denotes the original kernel matrix and  $\mathbf{k}_i$  denotes the  $i$ -th column vector of  $\mathbf{K}$ . Applying the convex TED to the derived RKHS leads to the following optimization problem:

$$\min_{\beta, \alpha_i \in \mathbb{R}^n} \sum_{i=1}^n \|\phi_M(\mathbf{x}_i) - \phi_M(\mathbf{X})\alpha_i\|^2 + \sum_{j=1}^m \frac{\alpha_{i,j}^2}{\beta_j} + \gamma \|\beta\|_1 \quad (5.63)$$

$$\text{subject to } \mathbf{x}_i \in \mathbf{X}, \beta_j \geq 0, j = 1, \dots, m$$

with the feature mapping  $\phi_M : \mathbb{R}^d \rightarrow \tilde{\mathcal{H}}$  from the original space to the RKHS. By setting  $\phi_M(\mathbf{X}) = [\phi_M(\mathbf{x}_1), \dots, \phi_M(\mathbf{x}_n)]$ , we have the following property between the feature mapping and the kernel matrix:  $\mathbf{K}_M = \phi_M(\mathbf{X})^T \phi_M(\mathbf{X})$ . After some derivations, we arrive at the following update rules, which are repeated until convergence [CH12]:

$$\beta_j \leftarrow \sqrt{\frac{1}{\gamma} \sum_{i=1}^n \alpha_{i,j}^2} \text{ for } j = 1, \dots, m \quad (5.64)$$

$$\alpha_i \leftarrow (\text{diag}(\beta)^{-1} + \mathbf{K}_M)^{-1} \mathbf{u}_i \text{ for } i = 1, \dots, n. \quad (5.65)$$

There,  $\text{diag}(\beta)$  denotes the diagonal matrix whose diagonal elements are set to the corresponding elements of  $\beta$  and  $\mathbf{u}_i$  is the  $i$ -th column vector of  $\mathbf{K}_M$ . After convergence, the data points are ranked according to  $\beta_j$ , ( $j = 1, \dots, m$ ) in descending order and the top  $m$  data points are selected.



### 5.2.4 Support vector machine active learning

The SVM<sub>Active</sub> [TC01] algorithm is directly coupled to the SVM classifier. The main idea is to select samples that minimize the version space of the classifier (i.e. the space of all possible classifier values  $\mathbf{w}$ ). In the SVM<sub>Active</sub> algorithm, we consider SVMs in the binary classification setting with the data  $\mathbf{X}$  and the labels  $\mathbf{Y}$ . For the binary classification problem,  $y_i \in \{-1, 1\}$ . For a general RKHS, the SVM classifier is of the form

$$f(\mathbf{x}) = \left( \sum_{i=1}^n \alpha_i \mathcal{K}(\mathbf{x}_i, \mathbf{x}) \right). \quad (5.66)$$

Additionally, if there exists a corresponding feature mapping  $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$ ,  $\mathcal{K}$  can be written as  $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u}) \cdot \phi(\mathbf{v})$  and  $f$  as:

$$f(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}), \text{ where } \mathbf{w} = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i). \quad (5.67)$$

Based on the introduced classifier, [TC01] proposes the SVM problem in the following form:

$$\begin{aligned} & \underset{\mathbf{w} \in \mathcal{H}}{\text{maximize}} && \min_i \{y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i))\} \\ & \text{subject to:} && \|\mathbf{w}\| = 1 \\ & && y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i)) > 0, i = 1, \dots, n \end{aligned} \quad (5.68)$$

and define the version space  $\mathcal{V}$  as the set of possible solutions  $\mathbf{w}$ , that satisfy the constraints:

$$\mathcal{V} = \{\mathbf{w} \in \mathcal{H} \mid \|\mathbf{w}\| = 1, y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i)) > 0, i = 1, \dots, n\}. \quad (5.69)$$

These definitions show a dual relationship between the classifier vectors  $\mathbf{w}$  and the feature vectors  $\phi(\mathbf{x})$ . If viewed in the space of  $\phi(\mathbf{x})$ , the vectors  $\mathbf{w}$  are hyperplane normal vectors, that separate the points  $\phi(\mathbf{x})$ , based on the condition  $y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i))$ . However, if viewed in the space of  $\mathbf{w}$ ,  $\phi(\mathbf{x})$  are hyperplane normal vectors, that constrain the possible values of points  $\mathbf{w}$ . Thus, in the space of  $\mathbf{w}$  the version space can be viewed as part of a sphere with radius 1, centered in the origin, that is constrained by hyperplanes based on the conditions  $y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i))$ . Additionally, the SVM optimization problem (5.68) now has the geometric interpretation of finding the center of the largest radius hypersphere, whose center can be placed in the version space and whose surface does not intersect with any of the constricting hyperplanes. Furthermore, the hyperplanes that are touched by the largest radius hypersphere, are exactly those hyperplanes, that correspond to the support vectors  $\phi(\mathbf{x}_i)$  in the dual space [TC01].

Based on these observations, Tong et al. [TC01] prove that an optimal active learning algorithm should choose samples that halve the area of the version space in each iteration. More specifically, for the SVM problem this involves selecting new samples for labeling in such a way their corresponding hyperplanes in the space of  $\mathbf{w}$  halve the area of the unit sphere, corresponding to the condition  $\|\mathbf{w}\| = 1$ . To address this question, Tong et al. [TC01] propose three different methods. The simple margin algorithm is based on the geometric interpretation that the unit vector  $\mathbf{w}$  obtained from the optimization is the center of the largest radius sphere that does not intersect any of the constricting hyperplanes. Thus,  $\mathbf{w}$  also lies approximately at the center of the version space and therefore selecting the sample whose hyperplane is closest to the the current vector  $\mathbf{w}$  corresponds to selecting the sample that is closest to bisecting the version space. This leads to the following rule, for selecting the next sample  $i^*$  for labeling in each iteration:

$$i^* = \arg \min_i |\mathbf{w} \cdot \phi(\mathbf{x}_i)|. \quad (5.70)$$

Additionally, the simple margin algorithm can be interpreted in the space of  $\phi(\mathbf{x})$  as selecting in each iteration the sample  $i$ , whose feature vector lies closest to the classifier hyperplane and therefore about which the classifier is less certain.

## 5.3 Proposed method

In this section, we describe our proposed active learning algorithm that uses a trace-norm regularized classifier as a training model and a visualization-based sample selection. Although this training model has been introduced recently, we further study its properties and compare it with the SVM.

### 5.3.1 Trace-norm regularized Classifier (TC)

The motivation for using a low-rank regularizer in classification comes from the observation that, when the SVM classifier is trained on large-scale data sets, the singular values of learning parameters (matrix  $\mathbf{W}$ ) have an exponential decay [Har+12]. By looking at the dimensions of  $\mathbf{W}$ , this can either be interpreted as the samples lying on a subspace of a lower dimension or as the classes being linear combinations of a smaller set of underlying prototype classes. Therefore, the authors of [Har+12] propose to leverage this property (i.e., the singular values of  $\mathbf{W}$  have an exponential decay) by minimizing the rank of the matrix  $\mathbf{W}$  and therefore keeping only the singular values with high magnitude.

We consider the set of  $n$  feature vectors  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of dimension  $d$  and the corresponding class labels  $\mathcal{Y} = \{y_1, \dots, y_n\}$  with a total number of  $k$  classes. The general linear multiclass classification problem involves learning a classifier

$g(\mathbf{x}) = \operatorname{argmax}_{l=1,\dots,k} \mathbf{w}_l^T \mathbf{x}$ , that predicts a class  $l$ , for each point  $\mathbf{x}$ . The classifier is specified by the  $k$  weight vectors  $\mathbf{w}_l$ . The general procedure to train the classifier involves minimizing the regularized empirical risk function:

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times k}} \lambda \Omega(\mathbf{W}) + \frac{1}{n} \sum_{i=1}^n L(\mathbf{W}; \mathbf{x}_i, y_i) \quad (5.71)$$

with the regularization penalty  $\Omega$ , the loss function  $L$  and the weight matrix  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_k]$ . One popular method for training the classifier is the OVR strategy [RK04], which involves splitting the problem into  $k$  binary classification problems, where for each class, the labels corresponding to this class are set to 1 and the labels corresponding to other classes to  $-1$ . The binary problems can then be solved by an SVM classifier [Vap82]. The first problem which arises when trying to apply the low-rank regularizer to the OVR optimization problem, is that each of the columns of  $\mathbf{W}$  is trained independently on the corresponding binary problem, while the low rank constraint requires the treatment of the matrix  $\mathbf{W}$  as a whole. Therefore, the authors of [Har+12] propose to use the multinomial logistic loss function, which treats all classes simultaneously. Introducing the low-rank enforcing penalty and the multinomial logistic loss function into (5.71), leads to the following objective function for minimization:

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times k}} \lambda_1 \operatorname{rank}(\mathbf{W}) + \lambda_2 \|\mathbf{W}\|_2^2 + R_n(\mathbf{W}) \quad (5.72)$$

where

$$R_n(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^n L(\mathbf{W}; \mathbf{x}_i, y_i) \quad (5.73)$$

and

$$L(\mathbf{W}; \mathbf{x}, y) = \log \left( 1 + \sum_{l \in \mathcal{Y} \setminus \{y\}} \exp \{ \mathbf{w}_l^T \mathbf{x} - \mathbf{w}_y^T \mathbf{x} \} \right). \quad (5.74)$$

This is a non-smooth non-convex optimization problem, which is difficult to solve. However, in [Har+12] a solution is provided, where the low-rank penalty is replaced by its convex surrogate, the trace-norm. The whole algorithm is summarized in Algorithm 2.

The algorithm so far has only been described for the linear case. Even though it is possible to introduce a high-dimensional mapping for data sets with nonlinear mappings, as shown for SVM, this method is not used here. The reason is that the trace-norm regularized classifier was specifically developed for real data sets with high-dimensional feature spaces. In state-of-the-art methods, high-dimensional image descriptors are used in combination with linear classifiers [Har+12].

**Algorithm 2** Solving trace-norm-regularized optimization problem (summary of algorithm in [Har+12])

---

**Input:** regularization parameters  $\lambda_1$  and  $\lambda_2$   
initial point  $\mathbf{W}_{\theta_0}$ , convergence threshold  $\epsilon$   
training points  $\mathcal{X}$  and labels  $\mathcal{Y}$

**Output:**  $\epsilon$ -optimal  $\mathbf{W}_\theta$

**Algorithm:**

**for**  $t=0,1,2,\dots$  **do**

  Compute top singular vector pair  $\{\mathbf{u}_t, \mathbf{v}_t\}$  of  $-\nabla \tilde{R}_n(\mathbf{W}_t)$

  Let  $g_t = \lambda_1 + (\nabla \tilde{R}_n(\mathbf{W}_t), \mathbf{u}_t \mathbf{v}_t^T)$

**if**  $g_t \leq -\epsilon/2$  **then**

$\mathbf{W}_{t+1} = \mathbf{W}_t + \delta \mathbf{u}_t \mathbf{v}_t^T$  with  $\delta$  found by line-search

$\theta_{t+1} = \delta$

$\theta_{t+1} = [\theta_t, \theta_{t+1}]$

**else**

    Check stopping conditions:

$\forall i \in \mathcal{I} : \frac{\partial \tilde{R}_n(\theta)}{\partial \theta_i} + \lambda_1 \geq -\epsilon$

$\forall i \in \mathcal{I} | \theta_i \neq 0 : \left| \frac{\partial \tilde{R}_n(\theta)}{\partial \theta_i} + \lambda_1 \right| \leq \epsilon$

**if** stopping conditions satisfied **then**

      stop and return  $\mathbf{W}_t$

**else**

      Compute  $\theta_{t+1}$  as a solution of the following restricted problem:

$$\min_{\theta_1, \dots, \theta_s} \lambda_1 \sum_{j=1}^s \theta_j + \tilde{R}_n \left( \sum_{j=1}^s \theta_j \mathbf{u}_j \mathbf{v}_j^T \right)$$

      subject to  $\theta_j \geq 0, j = 1, \dots, s$

**end if**

**end if**

**end for**

---

### 5.3.1.1 Comparison to the SVM algorithm

The introduced low-rank based classifier has some similarities to the SVM algorithm. Both estimate the classification boundary through the matrix  $\mathbf{W}$  by minimizing the regularized loss over the set of samples. However, there are also some important differences. These are: 1) the way multiclass problems are treated; 2) the type of loss functions; and 3) the regularizers used, which might lead to the existence of support vectors.

### Multiclass problem

The first difference is that the SVM algorithm cannot deal with multiple classes simultaneously and therefore has to resort to the (OVR) strategy, which involves solving multiple binary training problems simultaneously. One problem of this approach is that the resulting binary problems are imbalanced. By increasing the number of classes, the degree of imbalance increases. The second problem is that the matrix  $\mathbf{W}$  is not treated as a whole. Instead, each column of  $\mathbf{W}$  is estimated independently by a separate classifier. Thus, the matrix  $\mathbf{W}$  cannot be regularized as a whole and its columns might be imbalanced relative to each other.

### Loss function and regularizer

The next difference is the type of loss function and regularizers used by the two algorithms. The SVM algorithm is a constrained optimization problem with the hinge loss function [CS00]. The hinge loss for the vector  $\mathbf{w}$  and a sample  $\mathbf{x}_i$  with label  $y_i$  is defined as:

$$l_h(\mathbf{w}) = \max\{0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\}. \quad (5.75)$$

With the help of the hinge loss, the SVM optimization problem can be transformed into the following unconstrained optimization problem [CS00]:

$$\min_{\mathbf{w}} \lambda \|\mathbf{w}\|^2 + \frac{1}{n} \sum_{i=1}^n \max\{0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\}. \quad (5.76)$$

This makes it possible to highlight the differences between the two classifiers more clearly. The TC uses a weighted combination of the  $L_2$  norm and the trace-norm of  $\mathbf{W}$ . Additionally, as mentioned in the SVM, the  $L_2$  norm of each column of  $\mathbf{W}$  is minimized independently, while for the TC, the two norms are minimized as a whole. Regarding the loss function, the SVM uses the non-smooth hinge loss function for the binary problem of whether a sample belongs to that class or not. Instead, the TC uses the multiclass logistic loss function, which is smooth and considers the difference of a sample class to all other classes simultaneously.

### Support vectors

Another difference between the two classifiers is the existence of support vectors. In the SVM classifier, due to the non-smooth hinge loss function, only a subset of the vectors contributes to the training of the classifier. These vectors are the support vectors. The existence of support vectors makes it possible to reduce the training time of the SVM significantly, since the computations have to be performed only over this subset of vectors. In the TC, support vectors do not exist explicitly due

to the smooth multiclass logistic loss function. However, it might be desirable to find an approximation of support vectors for some applications, e.g. reduction of the number of samples for training. In the following, an approach for approximating the support vectors is presented.

The approach is based on the hinge loss function of the SVM classifier (5.75). Let  $y_i$  denote the true label of sample  $\mathbf{x}_i$ . Then for each column  $\mathbf{w}_l$  of  $\mathbf{W}$ , the hinge loss is computed by:

$$l_{h;l} = \max\{0, 1 - \hat{y}_i \mathbf{w}_l^T \mathbf{x}_i\}, \quad (5.77)$$

where  $\hat{y}_i = 1$  if  $l = y_i$ , and  $\hat{y}_i = -1$  otherwise. If any of the resulting hinge loss terms is greater than zero, i.e:

$$l_{h;l} \geq 0, \quad l = 1, \dots, k, \quad (5.78)$$

then sample  $\mathbf{x}_i$  is selected as a support vector of the TC. Figure 5.3 shows an example of the support vectors computed from a synthetic data set. The data set consists of three classes, with 100 samples per class. The positions of the samples are computed by Gaussian distributions. The support vectors are denoted by black crosses. Figure 5.3(a) shows the support vectors of the SVM classifier and Figure 5.3(b) shows the support vectors of the TC computed by inserting the columns of  $\mathbf{W}$  into the hinge loss function. In the SVM classifier, the support vectors come up due to the non-smooth loss function. The number of support vectors is controlled by the regularization parameter and with an increasing amount of support vectors over-fitting occurs. In the TC classifier, the smooth multiclass logistic loss function is used instead, which takes all samples into account and therefore by default has no support vectors. The usage of this loss function is required in order to use the trace-norm regularization term, which treats the matrix  $\mathbf{W}$  as a whole. This is not possible with the OVR approach and the usage of the SVM hinge loss, which treats the multiclass problem as multiple binary problems. However, even though the TC classifier takes all samples into account, the over-fitting problem is avoided with the added trace-norm regularization term. The disadvantage of taking all samples into account in the TC classifier is the increased computation time, however a higher accuracy is achieved through this. The support vectors shown in Figure 5.3(b) for the TC classifier are an approximation, which are estimated by introducing the hinge loss function into the matrix computed by the TC classifier and can be used in order to determine which vectors have the highest contribution to the optimization problem.

### Computational complexity

The computational complexity is an important factor for the comparison of the SVM to the TC. For the solution of the SVM optimization problem, many different optimization techniques have been developed over the years. Therefore, it is difficult

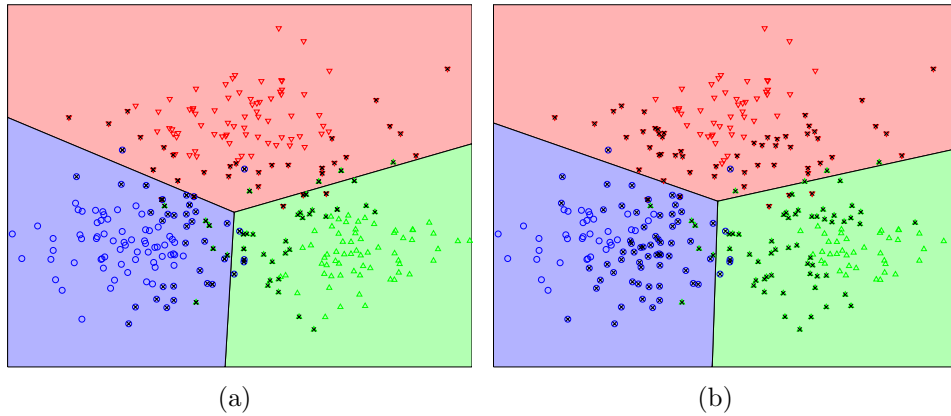


Figure 5.3: Visualization of support vectors for a synthetic dataset. Black crosses denote the support vectors (a) Support Vectors of SVM classifier (b) Support Vectors of the TC, approximated by hinge loss function. In the TC, more features are considered support vectors.

to estimate the overall computational complexity. For the *LibSVM* library [CL11], which is the SVM implementation used in this work, the overall computational complexity has been estimated to be  $O(ndp)$  for the binary problem, where  $n$  is the number of samples,  $d$  is the dimension of the feature vectors and  $p$  is the number of iterations. For the extension of the binary problem to the multiclass problem with  $k$  classes, the binary problem has to be solved  $k$  times in total. This leads to an overall time complexity of  $O(kndp)$ .

For the TC, the most expensive steps are the computation of  $\nabla \tilde{R}_n(\mathbf{W})$  and of the top singular vector pair of  $\nabla \tilde{R}_n(\mathbf{W})$ . The top singular vector pair can be computed by the Lanczos method [Che05], which has a time complexity of  $O(dk)$ . The computation of  $\nabla \tilde{R}_n(\mathbf{W})$  involves two steps, the matrix products  $\mathbf{w}_l^T \mathbf{x}_i$ , which have a computational complexity of  $O(ndk)$  and the summation of the logistic function over all terms, which has a computational complexity of  $O(nk)$ . Thus, the overall computational complexity of the TC is also on the order of  $O(kndq)$ , with  $q$  denoting the number of training iterations. It should be noted, however, that the TC requires more training iterations in total and more computations are performed during each training iteration. In this work, the training of the trace-norm regularized classifier was implemented in the scientific programming package Matlab. The most expensive calculations, like  $\nabla \tilde{R}_n(\mathbf{W})$ , have been implemented in C++ in the format of mex files. Table 5.1 summarizes the training complexities and the measured training times on a real data set for the different implementations. The real data set consists of 500 samples of the Synthetic Aperture Radar (SAR) data set, grouped in 15 classes. Each image is represented by the Bag-of-Word (BoW) model of SIFT local descriptors extracted from images.

Classifier	SVM	TC (Matlab)	TC (C++)
Time Complexity	$O(knd\mp)$	$O(kndq)$	$O(kndq)$
Runtimes [s]	0.262	386.228	23.220

Table 5.1: Computational complexity and actual training times of classifiers on SAR data set represented by BoW of SIFT feature descriptors.

### 5.3.2 Active learning with TC

The TC can be employed in an active learning framework as the training model. For instance, it can be based on iteratively training the classifier on the available subset of labeled samples and selecting the next sample for labeling as the point which is closest to the current boundary of the classifier, similar to the  $SVM_{Active}$  algorithm [TC01]. Let  $\mathcal{X}$  denote the set of all available samples and  $\mathcal{L}$  denotes the set of labeled samples. With the current weight matrix  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_k]$  the margin of each sample is:

$$\mu_i = \max_{l=1, \dots, k} \mathbf{w}_l^T \mathbf{x}_i. \quad (5.79)$$

The index of the next sample for labeling is then given by:

$$i_l = \operatorname{argmin}_i \{\mu_i | \mathbf{x}_i \in \mathcal{X} \setminus \mathcal{L}\}. \quad (5.80)$$

Training the classifier with each new labeled sample can be done efficiently by storing the matrix  $\mathbf{W}$  and setting it as the starting point for the next training iteration. The whole active learning algorithm is summarized in Algorithm 3.

### 5.3.3 Visualization-based sample selection

In this section, a novel active learning method is introduced, which is based on the principles of uncertainty and expected model change. Compared to the existing active learning algorithms, the difference here is that the sample selection strategy is also coupled to the visualization method in addition to the classifier. In the visualization, a ranked list of predictions ordered by confidence is shown to the user and the user is asked to select the first incorrectly predicted sample from one class. The algorithm can be used in combination with any classifier, for which a confidence metric can be computed, including the SVM and the TC introduced in the previous section.

For the description of the algorithm we consider the set of  $n$  samples with feature vectors  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of dimension  $d$  and the corresponding class labels  $\mathcal{Y} = \{y_1, \dots, y_n\}$  with a total number of  $k$  classes. Additionally, let  $\mathcal{L}$  denotes the labeled samples.



**Algorithm 3** Active learning with TC [Bab+15b]

**Input:** training points  $\mathcal{X}$  and labels  $\mathcal{Y}$   
initial set of labeled samples  $\mathcal{L}_0$   
total number of points to label  $m$

**Output:** new set of labeled samples  $\mathcal{L}$   
weight matrix  $\mathbf{W}$

**Algorithm:**

```

for  $t=0,1,2,\dots$  do
  obtain  $\mathbf{W}_t$  by applying Algorithm 2 on  $\mathcal{L}_t$  with initial weight matrix  $\mathbf{W}_{t-1}$ 
  if  $|\mathcal{L}_t| = m$  then
    stop and return  $\mathbf{W}_t, \mathcal{L}_t$ 
  end if
  Compute margin  $\mu_i$  for each sample according to (5.79)
  Select point  $\mathbf{x}_i$  with smallest margin according to (5.80) and set  $\mathcal{L}_{t+1} = \mathcal{L}_t \cup \{\mathbf{x}_i\}$ 
end for

```

The main idea of this algorithm is to select samples for labeling, which introduces the highest change into the model of a trained classifier. This is achieved by letting the algorithm predict labels during each iteration and asking the user to correct a label, which the algorithm is certain about, but predicts incorrectly. As the measure of certainty, we use the extension of the margin as suggested in the SVM<sub>Active</sub> algorithm [TC01] to a multiclass classifier. Given the current weight matrix  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_k]$  of the classifier, we define the margin  $\mu_i$  of each sample as

$$\mu_i = \max_{l=1,\dots,k} \mathbf{w}_l^T \mathbf{x}_i \quad (5.81)$$

and the predicted label  $\tilde{y}_i$  of each sample as

$$\tilde{y}_i = \arg \max_{l=1,\dots,k} \mathbf{w}_l^T \mathbf{x}_i. \quad (5.82)$$

Let  $I_{\tilde{l};i}$  denote the image of the unlabeled sample  $\mathbf{x}_{\tilde{l};i} \in \mathcal{X} \setminus \mathcal{L}$ , predicted with label  $\tilde{l}$ . Then the algorithm during each iteration arranges these images in a table with increasing margins for each class, i.e.

$$i \geq j \Leftrightarrow \mu_{\tilde{l};i} \geq \mu_{\tilde{l};j} \quad (5.83)$$

as suggested in Table 5.2 and lets the user select the first sample  $\mathbf{x}_{\tilde{l};m}$  in a class that is labeled incorrectly, i.e.

$$y_{\tilde{l};m} \neq \tilde{y}_{\tilde{l};m} \text{ and } y_{\tilde{l};i} = \tilde{y}_{\tilde{l};i} \text{ for } i = 1, \dots, m-1 \quad (5.84)$$

$\tilde{l} = 1$	$\tilde{l} = 2$	$\dots$	$\tilde{l} = k$
$I_{\tilde{l};1}$	$I_{\tilde{l};2}$		$I_{\tilde{l};k}$
$\vdots$	$\vdots$	$\dots$	$\vdots$
$I_{\tilde{l};n_1}$	$I_{\tilde{l};n_2}$		$I_{\tilde{l};n_k}$

Table 5.2: Predicted samples images presented to the user during each iteration

and relabel it. Since the samples are sorted with decreasing certainty, we can expect to achieve a big correction in the model by selecting the first incorrectly labeled sample. For the next iteration, the relabeled samples are added to the set of labeled samples  $\mathcal{L}$ .

$$\mathcal{L} = \mathcal{L} \cup \{x_{\tilde{l};1}, \dots, x_{\tilde{l};m}\}. \quad (5.85)$$

This is repeated for the desired number of iterations. Since the algorithm lets the user select a sample in the wrong class in each iteration, it is called First Certain Wrong Labeled (FCWL). The algorithm is summarized in Algorithm 4.

---

**Algorithm 4** FCWL: Active Learning with incorrect label correction by user

---

**Input:** training points  $X$  and labels  $Y$   
initial set of labeled samples  $L_0$   
total number of iterations  $p$

**Output:** new set of labeled samples  $L$

**Algorithm:**

**for**  $t = 0, 1, 2, \dots, p$  **do**

Obtain  $\mathbf{W}_t$  by training classifier on  $L_t$

Compute margin  $\mu_i$  for each sample according to (5.81)

Predict label  $\tilde{y}_i$  for each sample according to (5.82)

Present samples to user according to table 5.2 and equation (5.83)

Let user relabel first sample  $x_{\tilde{l};m}$  with incorrect predicted label from one class  
and set  $L_{t+1} = L_t \cup \{x_{\tilde{l};1}, \dots, x_{\tilde{l};m}\}$

**end for**

return  $L_t$

---

A schematic diagram of the FCWL algorithm for an optical data set is presented in Figure 5.4. On the top row we see images representing the different categories of the data set. Then, below them, the algorithm places the images based on their predicted labels in the corresponding categories. In this example, the algorithm already has many correct predictions in class 2 with the only incorrect prediction being the last one. So by selecting this image, the user can label all previous images from this category as correct and relabel the incorrect one.

In [Set10], active learning algorithms are categorized based on the sample



Figure 5.4: Example list of images presented by the algorithm for an optical dataset. The first row contains images representing each class. The sample images are arranged in the columns, which correspond to the predicted class, with decreasing margin [Bab+14b].

selection heuristic, such as uncertainty sampling, expected model change, query by committee and variance reduction. Our proposed active learning algorithm fits into the categories of uncertainty sampling and expected model change. However, the difference in which these principles are applied in this algorithm is that the samples are presented to the user in an ordered fashion and then the user selects which sample to label. This is contrary to previous active learning algorithms, where the samples for labeling are directly selected by the algorithm. Additionally, the expected model change of the classifier in the previous algorithms is estimated based on the gradient magnitude of the classifier optimization function for each sample. Here, the samples with the highest expected model change based on the correction introduced by the user are selected.

For the training of a classifier with the FCWL algorithm, a user interface was developed during this work. The user interface presents the list of ranked images to the user, as described in the previous section, and asks the user to select the first incorrectly predicted image from one category and relabel it. Additionally, some information about the training progress is given. A sample screenshot of the user interface for the SAR data set is presented in Figure 5.5. In this example, the training is currently at iteration 40 and 209 images have been labeled so far. Additionally, the plots show that the prediction accuracy on the test and training set is currently at about 50%.

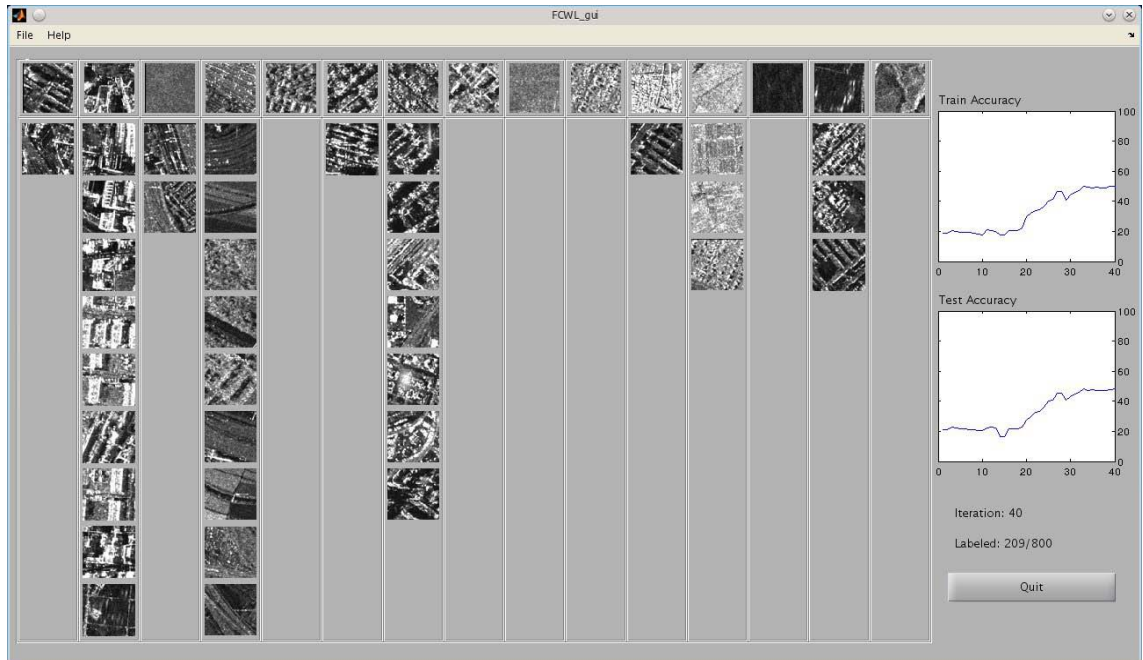


Figure 5.5: Snapshot of the user interface for the FCWL algorithm on SAR data set after 40 iterations

## 5.4 Experiments

After introducing two novel methods for active learning, we now turn to experiments conducted to compare the introduced methods to state-of-the-art algorithms. The data set used for evaluation is the SAR data set, from which different types of feature vectors are extracted. In the experiments, the accuracy of the different active learning algorithms will be presented for an increasing number of samples. Additionally, some results will be presented that show the effects of the different regularization parameters used in the algorithms.

### 5.4.1 Data sets

The SAR data set consists of a collection of 3434 images of the size  $160 \times 160$  pixels, which are grouped in 15 classes. The content of each image is represented by a feature vector computed by the Bag-of-Word (BoW) model different local descriptors, namely SIFT [Low04], BoW model of Weber Local Descriptor (WLD) [Jie+08] and Gabor [LW02]. Each feature vector is of length 64, which leads to a matrix of size  $3434 \times 64$ . Furthermore, the whole feature matrix is normalized to the range of  $[-1, 1]$  for each experiment. The SAR data set is fully introduced in Appendix A.

### 5.4.2 Setup

In addition to our proposed method, the following active learning methods were also applied to the data set.

- TED [YBT06], which defines a cost function based on the covariance of the prediction error of a least squares classifier. Then, in each iteration the sample that minimizes the value of the cost function is selected for labeling.
- MAED [CH12], which extends the TED algorithm with a manifold adaptive kernel in order to incorporate the manifold structure into the selection process.
- $LLR_{Active}$  [Zha+11], which minimizes the error of reconstructing the data set from the selected samples and the matrix describing the locally linear embedding.
- $SVM_{Active}$  [TC01], which iteratively adds points closest to the boundary of an SVM classifier to the training set and trains the classifier on the new set. To extend this algorithm to multiple classes, an OVR classifier is used and the margin of each point is computed based on its distance to the corresponding winning classifier.
- Random sampling method, which randomly selects a given number of points

For the compared Active Learning algorithms, the SVM with OVR scheme was used for training and classification. Several kernels such as Gaussian, Chi-square, and linear were used and the best results were achieved by using linear kernel. The metric used for comparison is the classification accuracy of the associated classifier. In order to obtain stable results, multiple tests were performed on different subsets of the data set and the average accuracy over all subsets computed. During each experiment, a random subset was chosen from the whole data set for training and testing. For the classifier parameter selection, we performed cross validation on each data set by increasing each parameter exponentially from the value  $10^{-4}$  to the value  $10^4$  and training the classifier for each parameter. Then, the parameter with the highest prediction accuracy was chosen for each data set. Similarly, we performed cross validation for the parameter selection of other active learning algorithms. Each active learning algorithm was applied to each data set for exponentially increasing parameter values between  $10^{-4}$  and  $10^4$  and 10 training samples. Then, the parameter with the highest prediction accuracy was chosen for the experiments.

### 5.4.3 Design 1: Active learning using TC

In this experiment, the performance of the TC-based active learning method is compared to the introduced algorithms. In addition to the proposed algorithm,

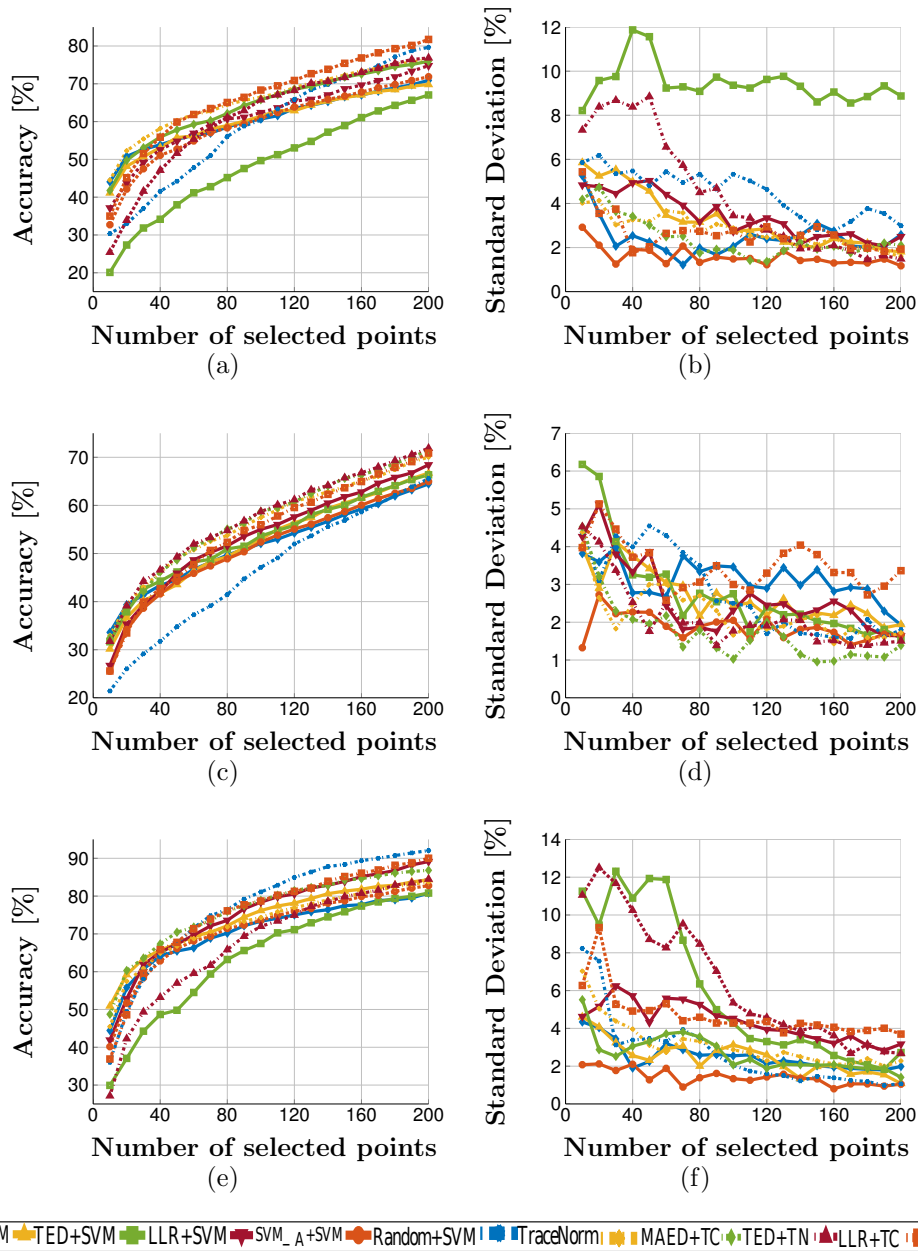


Figure 5.6: Classification results of three features representing the SAR data set. The first and second columns show the mean and the standard deviation of accuracy, respectively. The first, second, and third rows show the results of the Gabor, SIFT, and WLD features, respectively.

which selects samples based on their distance to the TC boundary, the TC is also applied to the samples selected by the other active learning algorithms.

The experiments were repeated 10 times for each data set and during each test a subset of 500 random samples was selected from the data set. Then each active learning algorithm selected an increasing number of samples from 20 to 200. For classification, the samples selected by each active learning algorithm were used as a training set and other samples of the subset were used as a test set. The classification results for all three features are presented in Figure 5.6.

The results show that the overall accuracy of the algorithms depends on the chosen feature descriptors. Best results are achieved with the WLD feature descriptor, where some algorithms achieve an accuracy of about 90%. Next comes the Gabor feature descriptor, which has on average 10% lower accuracy. Finally, there is the SIFT feature descriptor, which has 10% less average accuracy than Gabor. For all active learning algorithms, we notice that coupling the algorithm with the TC nearly always leads to a higher accuracy compared to coupling the algorithm with the SVM-classifier. The proposed active learning algorithm performs poorly on the SIFT feature descriptors, but improves in performance on the Gabor feature descriptors. It outperforms the other algorithms for an increasing number of samples on the WLD feature descriptors, which are the most important, since overall the highest accuracy is achieved here.

In general, we notice that algorithms based on the sample distribution, like  $LLR_{Active}$  and MAED, perform well for a small number of samples and that algorithms based on the available label information and trained classifier perform better as the number of samples increases. This is because in the beginning the trained classifier usually lies far away from the real classification boundary, and therefore the selected samples might actually be samples that do not contribute much information for training. However, as the number of samples increases and the classifier finds the position of the real boundary, the samples it selects for labeling have a high probability of becoming support vectors in the next training iteration. On the other hand, algorithms that select samples based on the sample distribution might achieve high accuracy in the beginning. These samples provide a good overall picture of how the data set is distributed in the feature space. However, as the number of training samples increases, selecting samples in this way provides less new information, since they are usually located further away from the class boundary in the feature space and therefore have a lower probability of becoming support vectors.

#### 5.4.3.1 TC parameter analysis

In order to analyze the behavior of the TC associated with the proposed active learning algorithm, experiments were also conducted on each data set with different values of  $\lambda_1$  and  $\lambda_2$ . For these experiments, 1000 samples from each data set were selected randomly as a test set and 100 randomly selected samples as a training set. Then, the TC was trained with different values of the parameters  $\lambda_1$  and  $\lambda_2$ . The results show that the parameter  $\lambda_2$  has less effect on the behavior and that in

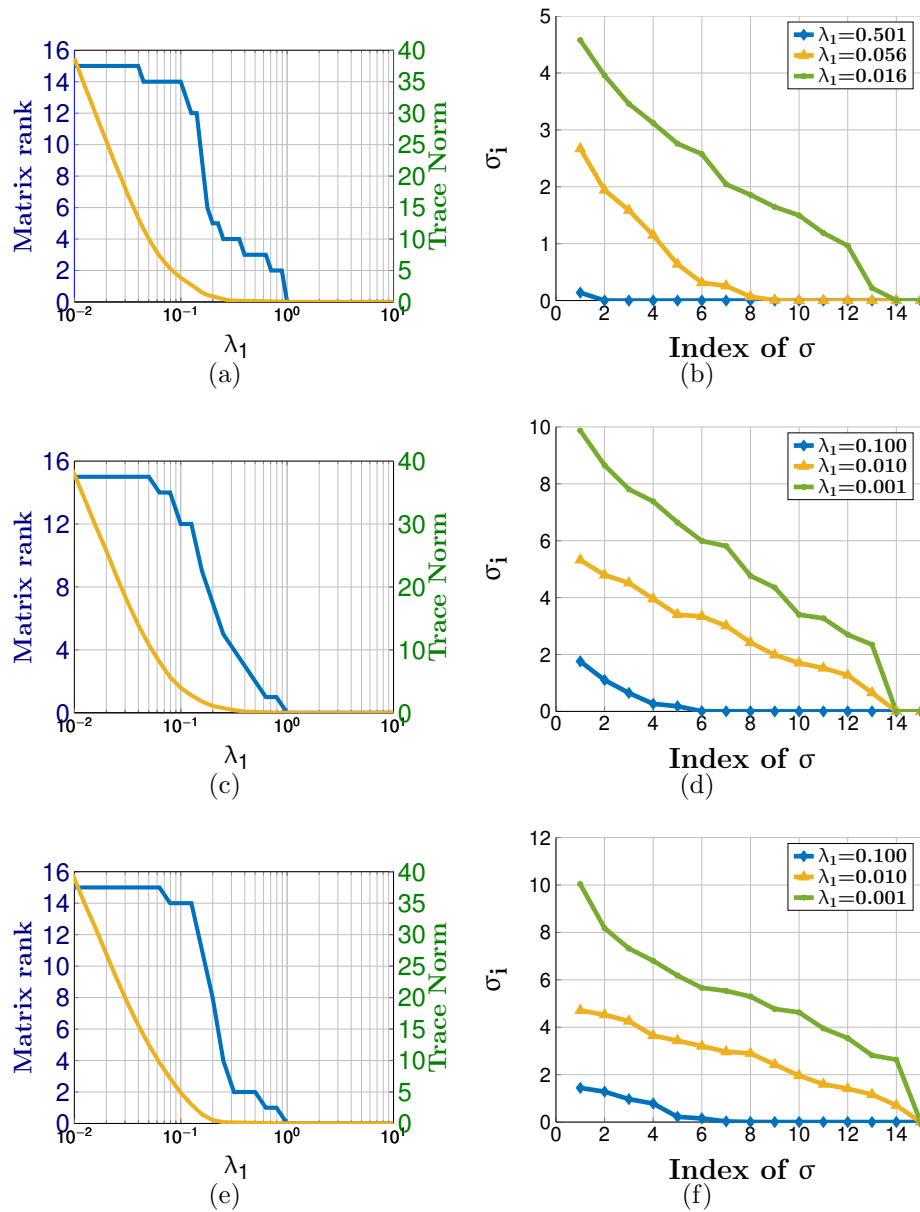


Figure 5.7: The left column shows the resulting matrix rank and trace-norm of the matrix  $\mathbf{W}$  for different values  $\lambda_1$  for different features. The right column shows the resulting singular values of  $\mathbf{W}$  for different  $\lambda_1$ . The first, second, and third rows are the results of the Gabor, SIFT, and WLD features, respectively.

general, the best results are achieved when  $\lambda_2$  scales similarly to  $\lambda_1$ . Therefore, in the figures showing the behavior of the classifier with respect to the parameters,  $\lambda_2$  was always chosen as  $\lambda_2 = 0.1\lambda_1$ .

Figure 5.7(a), Figure 5.7(c), and Figure 5.7(e) show the resulting matrix rank



and trace-norm of the matrix  $\mathbf{W}$  for different values of  $\lambda_1$  around the area where the matrix rank drops from the maximum rank to zero for Gabor, SIFT, and WLD features, respectively. The results show that an increase of the value of  $\lambda_1$  leads to a decrease in the trace-norm and therefore a decrease in the rank of the matrix from the maximum value to zero. Additionally, Figure 5.7(b), Figure 5.7(d), and Figure 5.7(f) show the resulting singular values of  $\mathbf{W}$  for different values of  $\lambda_1$ . Again, we see that as the value of  $\lambda_1$  increases, the average value of the singular values decreases and more singular values become zero.

#### 5.4.4 Design 2: Visualization-based active learning

Similar experiments were performed for the analysis of the interactive visualization-based active learning. The proposed active learning algorithm was applied in conjunction with the TC and SVM classifiers, while for the other algorithms, only the SVM classifier was used as classifier. The experiments were repeated again 10 times for each data set, but here in order to keep the results objective due to the multiple point selection of the proposed active learning classifier, different subsets were used for training and for testing. Specifically, during each experiment, a subset of 500 samples was selected as a training set and a different subset of 500 samples as a test set. Making the test set completely separated from the training set leads to a reduction in accuracy for all algorithms. The classification results are presented for the three features representing the SAR data set. Figure 5.8(a), Figure 5.8(c) and Figure 5.8(e) show the mean of classification accuracy for the Gabor, SIFT, and WLD features, respectively. Moreover, the standard deviations of classification accuracy are presented in Figure 5.8(b), Figure 5.8(d), and Figure 5.8(f) for the Gabor, SIFT, and WLD features, respectively. Again we see a dependence of the overall accuracy on the choice of feature descriptors with WLD leading to the highest accuracy and SIFT to the lowest. The plots show that the FCWL algorithm with the TC classifier outperforms the other active learning algorithms with all feature descriptors for an increasing number of interactions. The improved performance of the FCWL algorithm with TC compared to FCWL with SVM can be explained again by the ability of the TC to deal with a high dimension of the feature space and a high number of samples. This gets amplified even more in the case of FCWL, where multiple samples are selected per iteration, leading to an overall higher number of samples. Thus, the FCWL with TC can repeatedly make more accurate predictions, which lead to even more samples being labeled correctly and thus an even higher performance after training. This property becomes clear in the plots, where the FCWL with TC significantly outperforms the other algorithms, as the number of selected samples increases beyond 200. However, the effect of the increasing performance due to the increased accuracy and therefore higher amount of correctly predicted labels, can be observed with the FCWL and SVM classifier, that for 200 samples it has a high difference in performance, compared to the other algorithms.

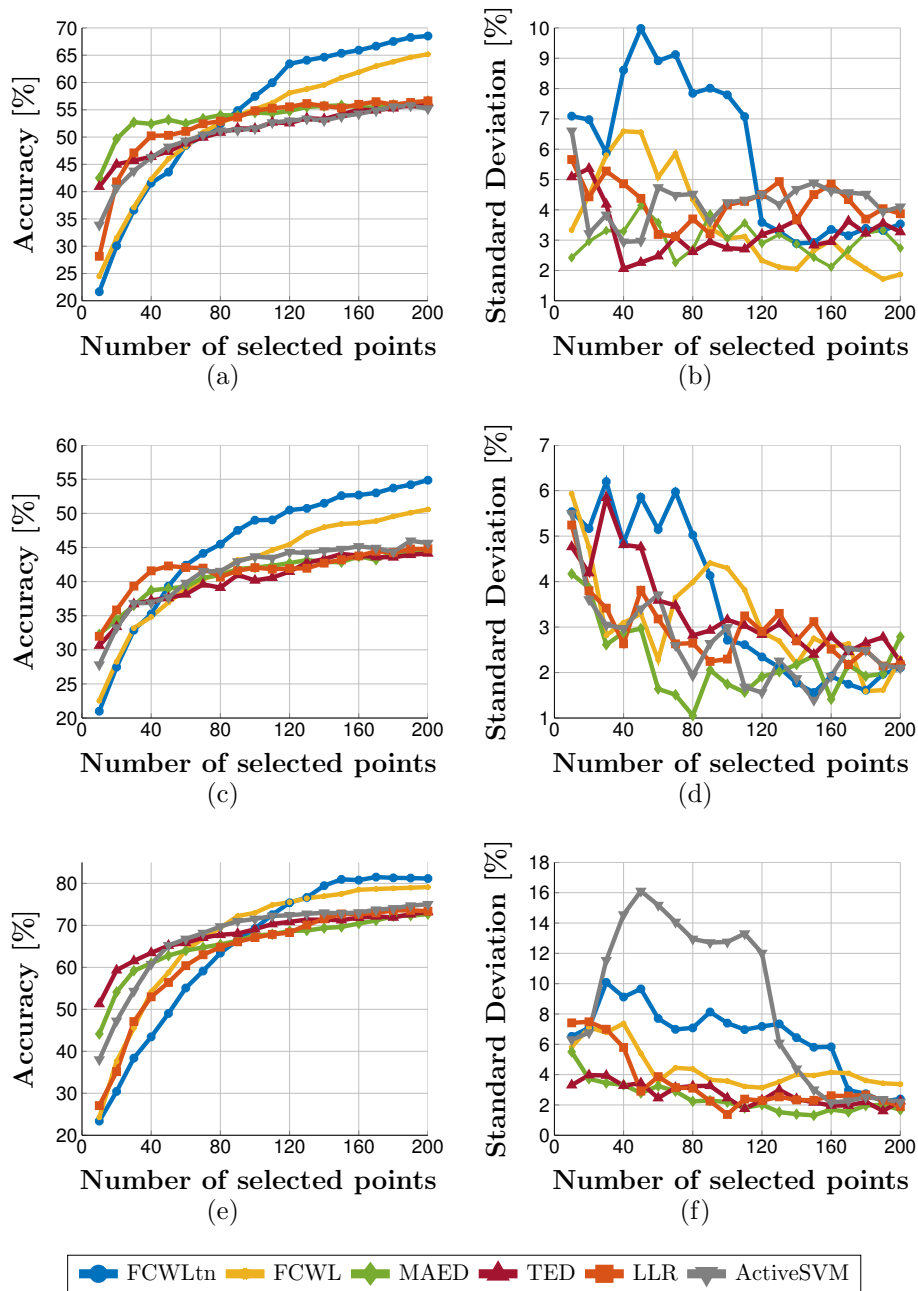


Figure 5.8: Classification accuracy of different active learning algorithms on SAR dataset. The first and second columns show the mean and the standard deviation of accuracy, respectively. The first, second and third rows show the results of Gabor, SIFT, and WLD features, respectively.

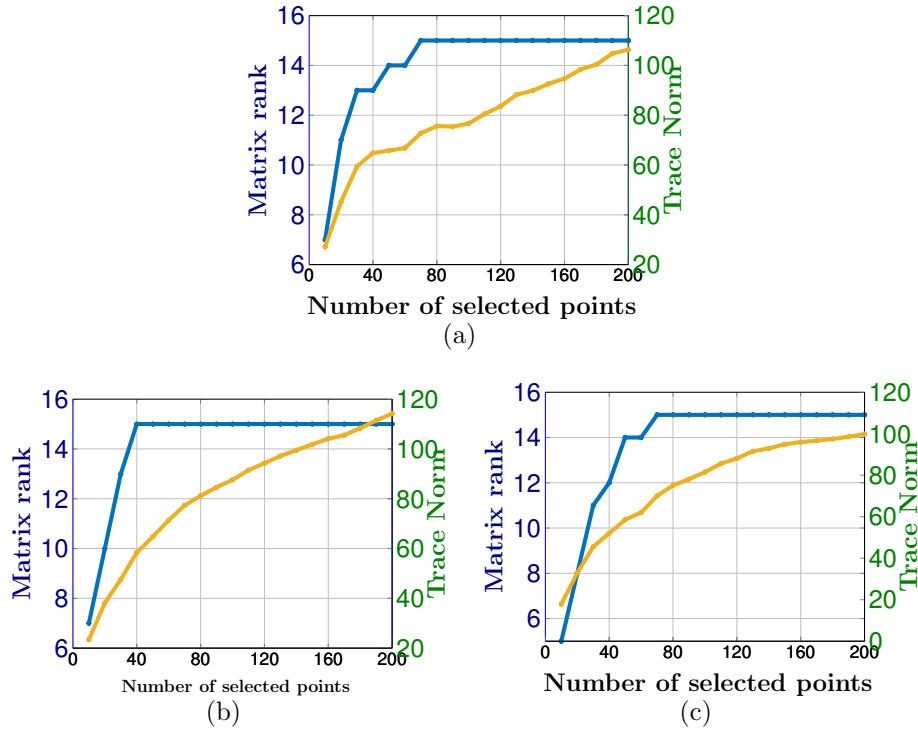


Figure 5.9: Matrix rank and trace-norm of FCWL algorithm with TC on SAR data represented by: (a) Gabor; (b) SIFT; and (c) WLD features.

The FCWL algorithm does worse than the other algorithms for a small number of samples. The reason for this can be found as before in the difference between classifiers and sample distribution based algorithms. When the number of samples is small, the predictions of the FCWL are still inaccurate and therefore the samples labeled by the user might not be the most informative. Additionally, it is possible that at the initial steps, the FCWL-based approaches know only a part of the classes and therefore do not predict some classes at all, which can lead to a further decrease in accuracy. However, as the number of samples increases, all classes can be predicted and the overall accuracy increases faster. On the other hand, algorithms like MAED and TED perform well at the beginning due to the selection of a more diverse set of samples that usually contains all classes but keep getting slower in accuracy later as a high sample diversity is no longer coupled to a high increase in classifier performance at this point.

Figure 5.9(a), Figure 5.9(b), and Figure 5.9(c) show the evolution of the matrix rank and trace-norm for the FCWL algorithm with trace-norm classifier and an increasing number of samples. These plots make the behavior of the FCWL algorithm with the trace-norm classifier more clear. At the beginning, we see a low matrix rank which means only a small subset of classes are known. The number of samples

required until all classes are identified varies for each feature descriptor. For example, for the Gabor descriptor all classes are known after 80 samples have been selected, for the SIFT descriptor 40 samples are necessary, and for the WLD descriptor 70. However, even at this point some of the new classes might still be represented only weakly. Therefore, the accuracy plots show that the FCWL algorithm with trace-norm classifier achieves good performance after the number of samples is beyond the point where all classes have been identified. The plots of the trace-norm value in Figure 5.9(a), Figure 5.9(b), and Figure 5.9(c) show how the trace-norm behaves as a relaxation of the matrix rank. It keeps increasing fast at the beginning as the rank of the matrix is growing and has a lower slope in the end when the rank of the matrix is constant.

## 5.5 Summary and conclusion

This chapter investigated active learning for the annotation and classification of image data sets. First, a solid background of active learning, including the concept and related work, was provided. We introduced a novel active learning algorithm by using a trace-norm regularized classifier as the training model and visualization-based sample selection as the sample selection strategy, in which by increasing the number of samples to label, the effects of overfitting can be reduced. The performed experiments on a SAR data set represented by three different features confirmed the quality of the proposed algorithm in comparison to other state-of-the-art techniques, where the proposed algorithm achieves the highest accuracy for an increasing number of samples. However, a disadvantage of the algorithm is the increasing computational effort required to solve the objective function with a coordinate descent algorithm. Developing more efficient methods to solve the optimization problem is therefore one possible direction for future work. Additionally, as the experimental results show, for a small number of selected samples the proposed algorithm is outperformed by other algorithms that select points based on the sample distribution. Therefore, another possible direction for future work can be combining the two approaches in order to develop an algorithm that takes sample distribution and label information into account and consistently provides high accuracy.

---

## Summary and Conclusion

This thesis has introduced a novel visual data mining system that mainly comprises interactive visualization and learning. Precisely, this work focuses on the intersection of machine learning and virtual reality and addresses multiple current issues in the area of human-machine communication for data mining applications. Most learning algorithms are based on the non-negative matrix factorization framework and the interactive visualization is based on the virtual reality technology. Virtual Reality was used to build up an immersive interactive 3D virtual environment for interactive data visualization. The main contributions of this work are: (1) discriminative data representation and dimensionality reduction based on non-negative matrix factorization and dictionary learning and the use of label or relative attributes information; (2) immersive interactive visualization of image collections and feature space; (3) interactive dimensionality reduction and data representation by introducing novel NMF based algorithms; (4) active learning for simultaneous annotation classifier learning. Each contribution has been fully introduced and discussed in a separate chapter.

In the following, the aforementioned contributions of this work are summarized and finally a concluding summary and an outlook are provided.

### 6.1 Summary

**Discriminative data representation** is introduced in Chapter 2. The main idea was to integrate label information in the matrix factorization process for dimensionality reduction and/or image (data) representation [Bab+ara]. A constrained optimization problem is presented where multiplicative update rules are proposed to find the solution to this problem. In addition to the label information, we showed that relative attributes can also be used as semantic information to generate discriminative features. At the end of this chapter, we proposed a novel relative attributes guided dictionary learning to generate discriminative sparse representation. We conducted

experiments to confirm the performance of this algorithm in comparison to other modern dictionary learning algorithms.

**Immersive visualization of image collections and feature space** is presented in Chapter 3. The first part of this chapter starts with a description of the immersive visualization system (i.e., Cave Automatic Virtual Environment (CAVE)). It provides information about the CAVE's components and how it is constructed. Then, the pipeline of data visualization, composed of feature extraction and dimensionality reduction, is discussed. Several examples of data visualization in the CAVE are provided. In the second part, a novel dimensionality reduction based on Non-negative Matrix Factorization (NMF) is introduced that takes into account the concerns of image visualization [Bab+arb]. This technique reduces the dimensionality and minimizes the occlusion among images, while preserving the structure of the data as much as possible. Experimental results show that this technique is flexible (by changing the controlling parameters) and optimal for data visualization.

**Interactive dimensionality reduction** is presented in Chapter 4. There, several algorithms were proposed that utilize the user's feedback from the CAVE, which is considered as a constraint in dimensionality reduction. All proposed methods are developed in the framework of non-negative matrix factorization [Bab+15d] and formulate the user's feedback as regularizers in the main objective function. The conducted experiments on both Earth Observation (EO) data and optical images confirm that the more feedback is used, the more discriminative property of the new feature is enhanced.

**Immersive active learning for the annotation of images** is an AL scenario proposed in Chapter 5. The three specifics of this system are: (1) the CAVE is used as the interface between the user and machine, where the results of the classifier (the distribution of images) are visualized; (2) the user decides which images should be selected for annotation and (3) a modern classifier (i.e., Trace-norm regularized classifier) is used as a training model [Bab+15b]. In contrast to other algorithms, where the machine selects images for annotation, in the proposed algorithm the user selects the images. The experimental results show that this approach outperforms the others.

## 6.2 Conclusion and outlook

Human-Machine communication for visual recognition and search is a challenging problem in pattern recognition and data mining. This thesis addresses this issue by introducing a novel immersive visual data mining system, in which immersive interactive data visualization and interactive learning algorithms play key roles. This

system comprises interactive dimensionality reduction, data representation and also active learning for the annotation of images.

Data representation and/or dimensionality reduction is a key step in every data mining application, where the content of the data is represented by a compact and informative feature vector. Therefore, interactive algorithms that are able to generate discriminative features are highly important in clustering and classification applications. In this thesis, we have used NMF as the basis of our proposed interactive dimensionality reduction and data representation algorithms. We found that NMF is a powerful computational tool in data representation and could be extended to generate customized features. However, it would also be interesting to investigate kernel learning techniques for dimensionality reduction in combination with the user interactions captured from the CAVE.

Visualization of image collections or feature space is essential for browsing and exploring image contents. Although dimensionality reduction techniques are widely used for data visualization, novel techniques are needed to consider display specifics such as the size and shape of display size. The proposed novel NMF-based technique aims at reducing the occlusion among images and shows excellent results based on the constraints defined by the user. However, for future work, it would be interesting to consider other constraints and extend the proposed technique to cover them. This would be possible by defining new regularizers that fulfill the required constraints and coupling them to the main objective function.

Active learning is a promising approach in annotating large amounts of unlabeled data. This approach aims to combine human and machine in classifier learning and data labeling. In this work, we proposed a novel approach that showed excellent results in comparison with state-of-the-art techniques. The proposed approach has great potential to assist the user in learning the classifier and decreasing the annotation time. It is envisioned that the proposed active learning algorithm can be further used in other applications such as activity recognition, human pose estimation, and natural language processing.

This thesis has investigated all the aforementioned challenges in detail and proposed several novel algorithms. We hope the community can benefit from this work. Although combining human and machine intelligence to solve data mining problems is still a long way off, the proposed algorithms may be used in future data mining applications.





Several data sets have been used in experimental results through this thesis. This appendix provides detailed information about the used data sets.

## A.1 Synthetic Aperture Radar (SAR)

SAR systems are imaging systems that illuminate the scene of interest with Electromagnetic (EM) radiation at microwave frequencies (300 MHz to 300 GHz) and measure the voltage returned from a scene of targets. The Radar Cross-Section (RCS) of the obtained voltage states how large a target is in a radar image. SAR systems are typically mounted on an aircraft and utilize the motion of a radar antenna over a target region in order to provide finer spatial resolution. Specifically, higher values for RCS show up bright targets, while lower values are dim. Targets with rough surfaces tend to scatter EM radiation back towards the radar. Thus, manmade targets such as buildings, vehicles, and roads have a high RCS. Since the ground is relatively flat, it scatters EM radiation away from the radar and its RCS is low [CL13]. SAR images are used in many different applications including surveillance, reconnaissance, foliage penetration, moving target indication, and environmental monitoring. SAR is often preferred over optical imaging systems since its performance is independent of daylight and visibility.

This data set<sup>1</sup> consists of 3434 SAR image patches with the size of  $100 \times 100$  pixels. There are several descriptors used to extract local feature from each patch. Using the BoW model, the extracted descriptors are coded into a single feature vector. The length of feature vector is equal to the number of visual words. The used descriptors are:

1. **Image intensity**- A sliding window with a size of 15 pixels is moving through the image and the image pixel intensities are concatenated to create a descriptor.

---

<sup>1</sup>The images are collected from TerraSAR-X data by Shiyong Cui, Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Germany

Next, using the BoW model, the computed descriptors from each image are coded to generate a single feature vector.

2. **Gabor**- A Gabor filter is applied to the images and a sliding window is moving through the image and then the Gabor coefficients in each window are concatenated to build up a descriptor. Next, using the BoW, the computed descriptors from each image are coded to generate a single feature vector.
3. **Weber**- A sliding window is moving through the image and the Weber Local Descriptors (WLD) are extracted and concatenated. The descriptors are coded using the BoW model to generate a single feature vector for each image.
4. **SIFT**- SIFT local descriptors are extracted from each image and then using the BoW model, the extracted descriptors are coded into a single feature vector.

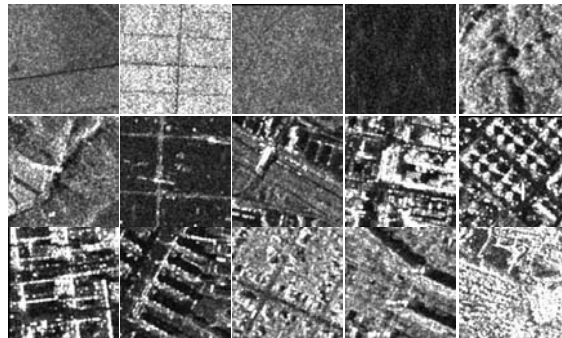


Figure A.1: Some sample images of the SAR image data set. Each image corresponds to one category.

## A.2 Caltech10

This data set contains the 10 biggest groups of the Caltech101 data set<sup>2</sup>, which is 3379 RGB-images. SIFT [Low04] descriptors were extracted from these images, then each image is represented by a 128-dimensional vector using the BoW model. In Figure A.2 some images of this data set are depicted.

## A.3 UC Merced Land Use

This data set contains 2100 images in 21 different categories with 100 images<sup>3</sup>. These categories are: agricultural, airplane, baseball-diamond, beach, buildings, chaparral,

<sup>2</sup>[http://www.vision.caltech.edu/Image\\_Datasets/Caltech101](http://www.vision.caltech.edu/Image_Datasets/Caltech101)

<sup>3</sup><http://vision.ucmerced.edu/datasets/landuse.html>

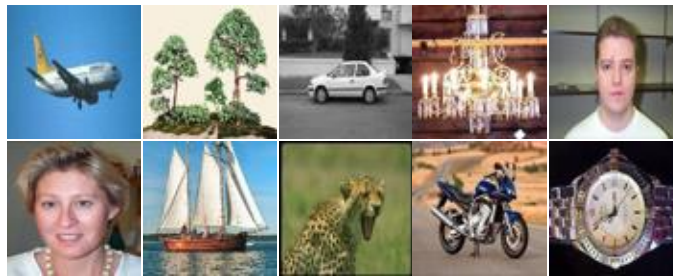


Figure A.2: Some sample images of the Caltech10 data set. Each image corresponds to one category.

dense-residential, forest, freeway, golf-course, harbor, intersection, medium-residential, mobile-home-park, overpass, parking-lot, river, runway, sparse-residential, storage-tanks, and tennis-court. From these images, SIFT descriptors are extracted and using the BoW model, each image is represented with a feature vector of length 64. In Figure A.3, some images of this data set are depicted.

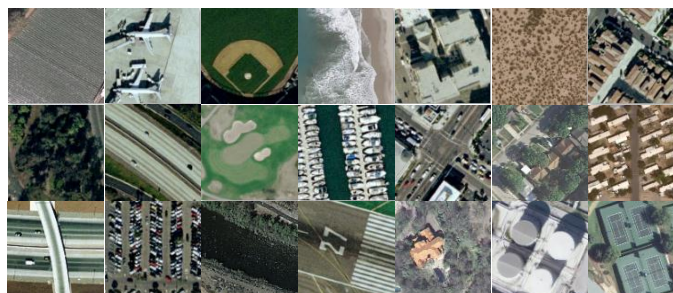


Figure A.3: The UC Merced Land Use is a manually labeled data set containing 21 classes of land-use scenes. Each image represents one sample of each group.

## A.4 Corel

This data set contains 1500 images from 15 categories <sup>4</sup>, which are: Africa, Beach, Bus, Card, Dyno, Elephant, Flower, Food, Grote, Horse, Mountain, Portrait, Rome, Sunset, and Tiger . First, the SIFT descriptors are extracted from each image. The extracted descriptors from each image are coded using the BoW model to create a 128-dimensional feature vector. Some sample images from this data set are depicted in Figure A.4

<sup>4</sup><https://archive.ics.uci.edu/ml/datasets/Corel+Image+Features>



Figure A.4: The Corel images data set. Each image represents one sample of each category.

## A.5 CMU PIE Faces

The CMU PIE face data set contains 3232 gray scale face images of 68 persons. Each person has 42 facial images under different light and illumination conditions. Each image is represented by the intensity values of its pixels that is a 1024-dimensional feature vector.



Figure A.5: Some sample images of the CMU PIE faces data set. Each image corresponds to one category.

## A.6 AT&T ORL Faces

The AT&T ORL data set<sup>5</sup> consists of 10 different images for each of 40 distinct subjects, thus 400 images in total. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling), and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position. In

<sup>5</sup><http://www.uk.research.att.com/facedatabase.html>

all experiments, images were preprocessed so that faces could be located. Original images were first normalized in scale and orientation such that the two eyes are aligned at the same position. Then, the facial areas were cropped into the final images for clustering. Each image is  $32 \times 32$  pixels with 256 gray levels per pixel.



Figure A.6: Some sample images of the ORL faces data set. Each image corresponds to one category.

## A.7 Yale Faces

The Yale faces data set <sup>6</sup> contains 165 gray-scale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration. These are: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. We do the same preprocessing for this data set as for the ORL faces data set. Thus, each image is also represented by a 1024-dimensional vector in image space.



Figure A.7: Some sample images of the PIE faces data set. Each image corresponds to one category.

<sup>6</sup><http://cvc.yale.edu/projects/yalefaces/yalefaces.html>

## A.8 Handwritten Digits

The Handwritten Digits data set<sup>7</sup> contains 10000 gray scale images of handwritten digits from 0-9 with 1000 images per class. The size of each image is  $16 \times 16$  pixels and hence its content is represented by a 256-dimensional feature vector.



Figure A.8: Some sample images of the Handwritten Digits data set. Each image corresponds to one category.

---

<sup>7</sup><http://www.cs.nyu.edu/~roweis/data.html>

# B

---

## Convergence Proofs

### B.1 Convergence Proof of DNMF

**Theorem B.1.1.** *The objective function in (2.8) is nonincreasing under the update rules (2.14), (2.15) and (2.16). The objective function is invariant under these update rules if and only if  $\mathbf{U}$ ,  $\mathbf{V}$  and  $\mathbf{A}$  are at a stationary point.*

We first note that the newly introduced terms don't depend on  $\mathbf{U}$ . Thus, the update rule for  $\mathbf{U}$  remains the same as in the original formulation [LS01] and Theorem B.1.1 is true for (2.14). For the convergence proof of the proposed update rule for  $\mathbf{V}$ , we follow a similar procedure as in [LS01]. We use an auxiliary function as the one introduced for the expectation maximization algorithm [DLR77]. The following property is true for an auxiliary function:

**Lemma B.1.2.** *If there exists an auxiliary function  $G$  for  $F(\mathbf{x})$  with the properties  $G(\mathbf{x}, \mathbf{x}') \geq F(\mathbf{x})$  and  $G(\mathbf{x}, \mathbf{x}) = F(\mathbf{x})$ , then  $F$  is non-increasing under the update*

$$\mathbf{x}^{t+1} = \arg \min_{\mathbf{x}} G(\mathbf{x}, \mathbf{x}'). \quad (\text{B.1})$$

*Proof.*  $F(\mathbf{x}^{t+1}) \leq G(\mathbf{x}^{t+1}, \mathbf{x}^t) \leq G(\mathbf{x}^t, \mathbf{x}^t) = F(\mathbf{x}^t)$  □

The equality  $F(\mathbf{x}^{t+1}) = F(\mathbf{x}^t)$  holds only if  $\mathbf{x}^t$  is a local minimum of  $F(\mathbf{x})$ . By iteratively applying update rule (B.1),  $\mathbf{x}$  converges to the local minimum of  $F(\mathbf{x})$ . We will now show that the update rule (2.15) for variable  $\mathbf{V}$  corresponds to minimizing an auxiliary function for the objective in (2.8). Since the update rule is essentially element wise, it is enough to show that the objective is non-increasing under the update for each element  $\mathbf{V}_{ab}$ . We first compute the derivatives of the objective function with respect to the variable  $\mathbf{V}_{ab}$ .

$$F'_{ab}(\mathbf{V}_{ab}) = (-2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U} - \alpha 2\mathbf{Q}^T\mathbf{A} + 2\alpha\mathbf{V}_l\mathbf{A}^T\mathbf{A})_{ab} \quad (\text{B.2})$$

$$\mathbf{F}_{ab}''(\mathbf{V}_{ab}) = \begin{cases} 2(\mathbf{U}^T\mathbf{U} + \alpha\mathbf{A}^T\mathbf{A})_{bb} & \text{if } a \leq N_l \\ 2(\mathbf{U}^T\mathbf{U})_{bb} & \text{otherwise.} \end{cases} \quad (\text{B.3})$$

Based on this, we introduce the auxiliary function as follows:

**Lemma B.1.3.** *Let  $\mathbf{F}_{ab}(\mathbf{V}_{ab})$  denote the part of  $\mathbf{F}$  relevant to  $\mathbf{V}_{ab}$ . Then, the function*

$$\begin{aligned} \mathbf{G}(\mathbf{V}_{ab}, \mathbf{V}_{ab}^t) = & \mathbf{F}_{ab}(\mathbf{V}_{ab}^t) + \mathbf{F}'_{ab}(\mathbf{V}_{ab}^t)(\mathbf{V} - \mathbf{V}^t)_{ab} \\ & + \frac{1}{\mathbf{V}_{ab}^t} [\mathbf{V}^t\mathbf{U}^T\mathbf{U} + \alpha(\mathbf{V}_l^t\mathbf{A}^T\mathbf{A})^+ + \alpha(\mathbf{Q}^T\mathbf{A})^-]_{ab} (\mathbf{V} - \mathbf{V}^t)_{ab}^2. \end{aligned} \quad (\text{B.4})$$

is an auxiliary function for  $\mathbf{F}_{ab}(\mathbf{V}_{ab})$ .

**Proof.** It is straightforward to check that  $\mathbf{G}(\mathbf{V}_{ab}^t, \mathbf{V}_{ab}^t) = \mathbf{F}(\mathbf{V}_{ab}^t)$ . For the condition  $\mathbf{G}(\mathbf{V}_{ab}, \mathbf{V}_{ab}^t) \geq \mathbf{F}(\mathbf{V}_{ab})$ , we compare the auxiliary function to the Taylor series expansion

$$\mathbf{F}_{ab}(\mathbf{V}) = \mathbf{F}_{ab}(\mathbf{V}^t) + \mathbf{F}'_{ab}(\mathbf{V}_{ab}^t)(\mathbf{V} - \mathbf{V}^t)_{ab} + \frac{1}{2}\mathbf{F}_{ab}''(\mathbf{V} - \mathbf{V}^t)_{ab}^2. \quad (\text{B.5})$$

Comparing the second order terms of the auxiliary function with the second order terms of the Taylor series expansion, we get the condition:

$$\begin{cases} \frac{1}{\mathbf{V}_{ab}^t} [\mathbf{V}^t\mathbf{U}^T\mathbf{U} + \alpha(\mathbf{V}_l^t\mathbf{A}^T\mathbf{A})^+ + \alpha(\mathbf{Q}^T\mathbf{A})^-]_{ab} \geq (\mathbf{U}^T\mathbf{U} + \alpha\mathbf{A}^T\mathbf{A})_{bb} & \text{if } a \leq N_l \\ \frac{1}{\mathbf{V}_{ab}^t} [\mathbf{V}^t\mathbf{U}^T\mathbf{U}]_{ab} \geq (\mathbf{U}^T\mathbf{U})_{bb} & \text{otherwise} \end{cases} \quad (\text{B.6})$$

We now check the inequality for each term on the left side of the equation with its corresponding term on the right side. For the NMF-term, we have for both cases:

$$\begin{aligned} (\mathbf{V}^t\mathbf{U}^T\mathbf{U})_{ab} &= \sum_c \mathbf{V}_{ac}^t (\mathbf{U}^T\mathbf{U})_{cb} = \mathbf{V}_{ab}^t (\mathbf{U}^T\mathbf{U})_{bb} + \sum_{c \neq b} \mathbf{V}_{ac}^t (\mathbf{U}^T\mathbf{U})_{cb} \\ &\Rightarrow \frac{(\mathbf{V}^t\mathbf{U}^T\mathbf{U})_{ab}}{\mathbf{V}_{ab}^t} \geq (\mathbf{U}^T\mathbf{U})_{bb}. \end{aligned} \quad (\text{B.7})$$

For the label-term in the case  $a \leq N_l$ , where  $\mathbf{V}_l = \mathbf{V}$  holds we have:

$$\begin{aligned} [(\mathbf{V}_l^t\mathbf{A}^T\mathbf{A})^+ + (\mathbf{Q}^T\mathbf{A})^-]_{ab} &\geq [\mathbf{V}_l^t\mathbf{A}^T\mathbf{A}]_{ab} = \sum_c \mathbf{V}_{l;ac}^t (\mathbf{A}^T\mathbf{A})_{cb} \\ &= \mathbf{V}_{l;ab}^t (\mathbf{A}^T\mathbf{A})_{bb} + \sum_{c \neq b} \mathbf{V}_{l;ac}^t (\mathbf{A}^T\mathbf{A})_{cb} \\ &\Rightarrow \frac{1}{\mathbf{V}_{ab}^t} [\alpha(\mathbf{V}_l^t\mathbf{A}^T\mathbf{A})^+ + \alpha(\mathbf{Q}^T\mathbf{A})^-]_{ab} \geq (\alpha\mathbf{A}^T\mathbf{A})_{bb}. \end{aligned} \quad (\text{B.8})$$



Since the inequality holds for all terms in both cases, it holds also for the sum of them and (B.6) is true. Thus, Lemma B.1.3 is true.  $\square$

***Proof of Theorem B.1.1.*** Inserting the auxiliary function (B.4) into (B.1) leads to the update rule (2.15). Thus, Theorem B.1.1 is true for (2.15). The convergence of the update rule (2.16) for  $\mathbf{A}$  follows directly by its definition. Since this update rule was derived by setting the derivative of the Lagrangian with respect to  $\mathbf{A}$  to 0 and  $\mathbf{A}$  is unconstrained, it follows from the convexity of the objective (2.9) in  $\mathbf{A}$  that this is equivalent to minimizing the objective with respect to  $\mathbf{A}$  in each iteration. Thus, Theorem B.1.1 is true for (2.16).  $\square$

## B.2 Convergence Proof of VISNMF

**Theorem B.2.1.** *The objective function in (3.20) is non-increasing under the update rules (3.26) and (3.27). The objective function is invariant under these update rules if and only if  $\mathbf{U}$ , and  $\mathbf{V}$  are at a stationary point.*

**Lemma B.2.2.** *If there exists an auxiliary function  $G$  for  $F(\mathbf{x})$  with the properties  $G(\mathbf{x}, \mathbf{x}') \geq F(\mathbf{x})$  and  $G(\mathbf{x}, \mathbf{x}) = F(\mathbf{x})$ , then  $F$  is non-increasing under the update*

$$\mathbf{x}^{t+1} = \arg \min_{\mathbf{x}} G(\mathbf{x}, \mathbf{x}'). \quad (\text{B.9})$$

**Proof.**  $F(\mathbf{x}^{t+1}) \leq G(\mathbf{x}^{t+1}, \mathbf{x}^t) \leq G(\mathbf{x}^t, \mathbf{x}^t) = F(\mathbf{x}^t)$  □

The equality  $F(\mathbf{x}^{t+1}) = F(\mathbf{x}^t)$  holds only if  $\mathbf{x}^t$  is a local minimum of  $F(\mathbf{x})$ . By iteratively applying update rule (B.9),  $\mathbf{x}$  converges to the local minimum of  $F(\mathbf{x})$ . We will now show that the update rule (3.27) for variable  $\mathbf{V}$  corresponds to minimizing an auxiliary function for the objective in (3.20). Since the update rule is essentially element wise, it is enough to show that the objective is non-increasing under the update for each element  $\mathbf{V}_{ab}$ . We first compute the derivatives of the objective function with respect to the variable  $\mathbf{V}_{ab}$ .

$$\begin{aligned} F(\mathbf{V}) = & -2\text{Tr}(\mathbf{X}\mathbf{V}\mathbf{U}^T) + \text{Tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) + \lambda_1 \text{Tr}(\mathbf{V}^T \tilde{\mathbf{L}}\mathbf{V}) \\ & + \lambda_1 \exp[-\beta \text{Tr}(\mathbf{V}^T \mathbf{L}^{(f)}\mathbf{V})] + \lambda_2 \log\left(\frac{1}{N^2} \sum_{i,j} G_{ij}\right) \end{aligned} \quad (\text{B.10})$$

$$\begin{aligned} F'_{ab}(\mathbf{V}) = & (-2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} + \lambda_1 (2\tilde{\mathbf{L}}\mathbf{V})_{ab} \\ & - \lambda_1 2\beta \exp[-\beta \text{Tr}(\mathbf{V}^T \mathbf{L}^{(f)}\mathbf{V})] (\mathbf{L}^{(f)}\mathbf{V})_{ab} \\ & + \lambda_2 \frac{2}{\sigma^2 \phi} \sum_i [G_{ia} (\mathbf{V}_{ib} - \mathbf{V}_{ab})] \end{aligned} \quad (\text{B.11})$$

$$\begin{aligned} F''_{ab}(\mathbf{V}) = & 2(\mathbf{U}^T\mathbf{U})_{bb} + \lambda_1 2\tilde{\mathbf{L}}_{aa} \\ & + \lambda_1 2 \left[ 2\beta^2 (\mathbf{L}^{(f)}\mathbf{V})_{ab}^2 - \beta \mathbf{L}_{aa}^{(f)} \right] \exp[-\beta \text{Tr}(\mathbf{V}^T \mathbf{L}^{(f)}\mathbf{V})] \\ & + \frac{2\lambda_2}{\sigma^2 \phi} \left( 1 + \frac{1}{\sigma^2} \sum_i G_{ia} (\mathbf{V}_{ib} - \mathbf{V}_{ab})^2 - \sum_i G_{ia} - \frac{2}{\sigma^2 \phi} \sum_i G_{ia} (\mathbf{V}_{ib} - \mathbf{V}_{ab})^2 \right) \end{aligned} \quad (\text{B.12})$$

Based on the computed derivatives, we introduce the following auxiliary function:

**Lemma B.2.3.** Let  $F_{ab}(\mathbf{V}_{ab})$  denote the part of  $F$  relevant to  $\mathbf{V}_{ab}$ . Then, the function

$$\begin{aligned} G(\mathbf{V}, \mathbf{V}^t) = & F_{ab}(\mathbf{V}^t) + F'_{ab}(\mathbf{V}^t)(\mathbf{V} - \mathbf{V}^t)_{ab} \\ & + \frac{(\mathbf{V} - \mathbf{V}^t)_{ab}^2}{\mathbf{V}_{ab}^t} \left\{ (\mathbf{V}^t \mathbf{U}^T \mathbf{U})_{ab} + \lambda_1 (\tilde{\mathbf{L}}^+ \mathbf{V}^t)_{ab} \right. \\ & \left. + \lambda_1 \beta \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right] (\mathbf{L}^{(f)-} \mathbf{V}^t)_{ab} + \frac{\lambda_2}{\sigma^2 \phi} \sum_i (\mathbf{G}_{ia} \mathbf{V}_{ib}^t) \right\}. \end{aligned} \quad (\text{B.13})$$

is an auxiliary function for  $F_{ab}(\mathbf{V}_{ab})$ .

**Proof.** It is straightforward to check that  $G(\mathbf{V}^t, \mathbf{V}^t) = F(\mathbf{V}^t)$ . For the condition  $G(\mathbf{V}, \mathbf{V}^t) \geq F(\mathbf{V})$  we compare the auxiliary function to the Taylor series expansion

$$F_{ab}(\mathbf{V}) = F_{ab}(\mathbf{V}^t) + F'_{ab}(\mathbf{V}^t)(\mathbf{V} - \mathbf{V}^t)_{ab} + \frac{1}{2} F''_{ab}(\mathbf{V}^t)(\mathbf{V} - \mathbf{V}^t)_{ab}^2 + O(\mathbf{V}_{ab}^3). \quad (\text{B.14})$$

Comparing the second order terms of the auxiliary function with the second order terms of the Taylor series expansion we get the condition:

$$\begin{aligned} & \frac{(\mathbf{V}^t \mathbf{U}^T \mathbf{U})_{ab}}{\mathbf{V}_{ab}^t} + \frac{\lambda_1 (\tilde{\mathbf{L}}^+ \mathbf{V}^t)_{ab}}{\mathbf{V}_{ab}^t} \\ & + \frac{\lambda_1 \beta (\mathbf{L}^{(f)-} \mathbf{V}^t)_{ab} \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right]}{\mathbf{V}_{ab}^t} + \frac{\lambda_2 \sum_i (\mathbf{G}_{ia} \mathbf{V}_{ib}^t)}{\sigma^2 \phi \mathbf{V}_{ab}^t} \\ & \geq (\mathbf{U}^T \mathbf{U})_{bb} + \lambda_1 \tilde{\mathbf{L}}_{aa} \\ & + \lambda_1 \beta \left[ 2\beta (\mathbf{L}^{(f)} \mathbf{V}^t)_{ab}^2 - \mathbf{L}_{aa}^{(f)} \right] \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right] \\ & + \frac{\lambda_2}{\sigma^2 \phi} \left[ 1 + \frac{1}{\sigma^2} \sum_i \mathbf{G}_{ia} (\mathbf{V}_{ib}^t - \mathbf{V}_{ab}^t)^2 - \sum_i \mathbf{G}_{ia} - \frac{2}{\sigma^2 \phi} \left( \sum_i \mathbf{G}_{ia} (\mathbf{V}_{ib}^t - \mathbf{V}_{ab}^t)^2 \right) \right]. \end{aligned} \quad (\text{B.15})$$

We now check the inequality for each term on the left side of the equation with its corresponding term on the right side. For the NMF-term we have:

$$\begin{aligned} (\mathbf{V}^t \mathbf{U}^T \mathbf{U})_{ab} & = \sum_c \mathbf{V}_{ac}^t (\mathbf{U}^T \mathbf{U})_{cb} = \mathbf{V}_{ab}^t (\mathbf{U}^T \mathbf{U})_{bb} + \sum_{c \neq b} \mathbf{V}_{ac}^t (\mathbf{U}^T \mathbf{U})_{cb} \\ & \Rightarrow \frac{(\mathbf{V}^t \mathbf{U}^T \mathbf{U})_{ab}}{\mathbf{V}_{ab}^t} \geq (\mathbf{U}^T \mathbf{U})_{bb}. \end{aligned} \quad (\text{B.16})$$

For the similarity-preserving / overview- term:

$$\begin{aligned} (\tilde{\mathbf{L}}^+ \mathbf{V}^t)_{ab} & \geq (\tilde{\mathbf{L}}^+ \mathbf{V}^t)_{ab} - (\tilde{\mathbf{L}}^- \mathbf{V}^t)_{ab} = (\tilde{\mathbf{L}} \mathbf{V}^t)_{ab} = \tilde{\mathbf{L}}_{aa} \mathbf{V}_{ab}^t + \sum_{c \neq b} \tilde{\mathbf{L}}_{ac} \mathbf{V}_{cb}^t \\ & \Rightarrow \lambda_1 \frac{(\tilde{\mathbf{L}}^+ \mathbf{V}^t)_{ab}}{\mathbf{V}_{ab}^t} \geq \lambda_1 \tilde{\mathbf{L}}_{aa}. \end{aligned} \quad (\text{B.17})$$

The inequality of the farness preserving term depends on the parameter  $\beta$ . For decreasing  $\beta$ , we have:

$$\begin{aligned}
 \frac{(\mathbf{L}^{(f)} - \mathbf{V}^t)_{ab}}{\mathbf{V}_{ab}^t} &\geq 2\beta (\mathbf{L}^{(f)} \mathbf{V}^t)_{ab}^2 - \mathbf{L}_{aa}^{(f)} \\
 \Rightarrow \lambda_1 \beta \frac{(\mathbf{L}^{(f)} - \mathbf{V}^t)_{ab} \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right]}{\mathbf{V}_{ab}^t} & \\
 &\geq \lambda_1 \beta \left[ 2\beta (\mathbf{L}^{(f)} \mathbf{V}^t)_{ab}^2 - \mathbf{L}_{aa}^{(f)} \right] \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right].
 \end{aligned} \tag{B.18}$$

For the entropy term, we have:

$$\frac{\sum_i (G_{ia} \mathbf{V}_{ib}^t)}{\mathbf{V}_{ab}^t} = \frac{G_{aa} \mathbf{V}_{ab}^t + \sum_{i \neq a} (G_{ia} \mathbf{V}_{ib}^t)}{\mathbf{V}_{ab}^t} \geq 1. \tag{B.19}$$

Furthermore, for increasing  $\sigma$  this can be expanded to

$$\begin{aligned}
 \frac{\sum_i (G_{ia} \mathbf{V}_{ib}^t)}{\mathbf{V}_{ab}^t} &\geq 1 + \frac{1}{\sigma^2} \sum_i G_{ia} (\mathbf{V}_{ib}^t - \mathbf{V}_{ab}^t)^2 \\
 \Rightarrow \frac{\lambda_2 \sum_i (G_{ia} \mathbf{V}_{ib}^t)}{\sigma^2 \phi \mathbf{V}_{ab}^t} & \\
 \frac{\lambda_2}{\sigma^2 \phi} \left[ 1 + \frac{1}{\sigma^2} \sum_i G_{ia} (\mathbf{V}_{ib}^t - \mathbf{V}_{ab}^t)^2 - \sum_i G_{ia} - \frac{2}{\sigma^2 \phi} \left( \sum_i G_{ia} (\mathbf{V}_{ib}^t - \mathbf{V}_{ab}^t)^2 \right) \right] & \\
 & \tag{B.20}
 \end{aligned}$$

Since the inequality holds for each term of the equation for the right choice of parameters  $\beta$  and  $\sigma$ , it does also hold for the sum of all terms and (B.15) is true. For the higher order terms of the Taylor series expansion we note, that the derivatives disappear for the NMF- and similarity-preserving- / overview- term and that all derivatives of the farness-preserving- and entropy-term are scaled by the factors  $\beta$  or  $1/\sigma^2$ , respectively. Therefore, for decreasing  $\beta$  and increasing  $\sigma$  those also become negligible and the condition  $G(\mathbf{V}, \mathbf{V}^t) \geq F(\mathbf{V})$  is true. Experimental results have shown that the values of parameters  $\beta$  and  $\sigma$ , required for good results, lie within the range where the algorithm converges.  $\square$

**Proof of Theorem B.2.1.** Inserting the auxiliary function (B.13) into (B.9) leads to the update rule (3.27). Thus, Theorem B.2.1 is true for (3.27).  $\square$

### B.3 Convergence Proof of CMNMF

**Theorem B.3.1.** *The objective function in (4.15) is non-increasing under the update rules (4.20) and (4.21). The objective function is invariant under these update rules if and only if  $\mathbf{U}$ , and  $\mathbf{Z}$  are at a stationary point.*

**Lemma B.3.2.** *If there exists an auxiliary function  $G$  for  $F(\mathbf{x})$  with the properties  $G(\mathbf{x}, \mathbf{x}') \geq F(\mathbf{x})$  and  $G(\mathbf{x}, \mathbf{x}) = F(\mathbf{x})$ , then  $F$  is non-increasing under the update*

$$\mathbf{x}^{t+1} = \arg \min_{\mathbf{x}} G(\mathbf{x}, \mathbf{x}'). \quad (\text{B.21})$$

**Proof.**  $F(\mathbf{x}^{t+1}) \leq G(\mathbf{x}^{t+1}, \mathbf{x}^t) \leq G(\mathbf{x}^t, \mathbf{x}^t) = F(\mathbf{x}^t)$  □

The equality  $F(\mathbf{x}^{t+1}) = F(\mathbf{x}^t)$  holds only if  $\mathbf{x}^t$  is a local minimum of  $F(\mathbf{x})$ . By iteratively applying update rule (B.21),  $\mathbf{x}$  converges to the local minimum of  $F(\mathbf{x})$ . We will now show that the update rule (4.21) for variable  $\mathbf{V}$  corresponds to minimizing an auxiliary function for the objective in (4.16). Since the update rule is essentially element wise, it is enough to show that the objective is non-increasing under the update for each element  $\mathbf{Z}_{ab}$ . We first compute the derivatives of the objective function with respect to the variable  $\mathbf{V}_{ab}$ .

$$F_{ab}(\mathbf{Z}) = F_{ab}(\mathbf{Z}_{ab}^t) + F'_{ab}(\mathbf{Z} - \mathbf{Z}_{ab}^t) + \frac{1}{2}F''_{ab}(\mathbf{Z} - \mathbf{Z}_{ab}^t)^2 \quad (\text{B.22})$$

$$F'_{ab}(\mathbf{Z}) = (-2\mathbf{W}^T \mathbf{X}^T \mathbf{U} + 2\mathbf{W}^T \mathbf{W} \mathbf{U}^T \mathbf{U})_{ab} \quad (\text{B.23})$$

$$F''_{ab}(\mathbf{Z}) = 2(\mathbf{W}^T \mathbf{W})_{aa} (\mathbf{U}^T \mathbf{U})_{bb} \quad (\text{B.24})$$

Based on the computed derivatives, we introduce the following auxiliary function

**Lemma B.3.3.** *Let  $F_{ab}(\mathbf{Z}_{ab})$  denote the part of  $F$  relevant to  $\mathbf{Z}_{ab}$ . Then, the function*

$$\begin{aligned} G(\mathbf{Z}, \mathbf{Z}_{ab}^t) &= F_{ab}(\mathbf{Z}_{ab}^t) + F'_{ab}(\mathbf{Z}_{ab}^t)(\mathbf{Z} - \mathbf{Z}_{ab}^t) \\ &\quad + \frac{(\mathbf{W}^T \mathbf{W} \mathbf{Z} \mathbf{U}^T \mathbf{U})_{ab}}{\mathbf{Z}_{ab}^t} (\mathbf{Z} - \mathbf{Z}_{ab}^t)^2 \end{aligned} \quad (\text{B.25})$$

*is an auxiliary function for  $F_{ab}(\mathbf{Z}_{ab})$ .*

**Proof.** Obviously,  $G(\mathbf{Z}^t, \mathbf{Z}^t) = F(\mathbf{Z}^t)$ . In order to show  $G(\mathbf{Z}, \mathbf{Z}^t) \geq F(\mathbf{Z})$ , we use the Taylor series expansion

$$F_{ab}(\mathbf{Z}) = F_{ab}(\mathbf{Z}_{ab}^t) + F'_{ab}(\mathbf{Z} - \mathbf{Z}_{ab}^t) + \frac{2(\mathbf{W}^T \mathbf{W})_{aa} (\mathbf{U}^T \mathbf{U})_{bb}}{2} (\mathbf{Z} - \mathbf{Z}_{ab}^t)^2 \quad (\text{B.26})$$

By comparing (B.25) and (B.26), we can see that  $G(\mathbf{Z}, \mathbf{Z}_{ab}^t) \geq F_{ab}(\mathbf{Z})$  is equal to show

$$\frac{(\mathbf{W}^T \mathbf{W} \mathbf{Z} \mathbf{U}^T \mathbf{U})_{ab}}{\mathbf{Z}_{ab}^t} \geq (\mathbf{W}^T \mathbf{W})_{aa} (\mathbf{U}^T \mathbf{U})_{bb}. \quad (\text{B.27})$$

By expanding the left side of the above inequality, we come up with

$$\begin{aligned} \frac{(\mathbf{W}^T \mathbf{W} \mathbf{Z} \mathbf{U}^T \mathbf{U})_{ab}}{\mathbf{Z}_{ab}^t} &= \frac{\sum_{l=1}^k (\mathbf{W}^T \mathbf{W} \mathbf{Z})_{al} (\mathbf{U}^T \mathbf{U})_{lb}}{\mathbf{Z}_{ab}^t} \\ &\geq \frac{(\mathbf{W}^T \mathbf{W} \mathbf{Z})_{ab} (\mathbf{U}^T \mathbf{U})_{bb}}{\mathbf{Z}_{ab}^t} \\ &\geq \frac{\sum_{l=1}^k (\mathbf{W}^T \mathbf{W})_{al} \mathbf{Z}_{lb}^t (\mathbf{U}^T \mathbf{U})_{bb}}{\mathbf{Z}_{ab}^t} \\ &\geq \frac{(\mathbf{W}^T \mathbf{W})_{aa} \mathbf{Z}_{ab}^t (\mathbf{U}^T \mathbf{U})_{bb}}{\mathbf{Z}_{ab}^t} \\ &= (\mathbf{W}^T \mathbf{W})_{aa} (\mathbf{U}^T \mathbf{U})_{bb} \end{aligned} \quad (\text{B.28})$$

□

**Proof of Theorem B.3.1.** Substitute the  $G(\mathbf{Z}, \mathbf{Z}^t)$  in (B.30) by (B.25), we have the optimization rule

$$\mathbf{Z}_{ab}^{(t+1)} = \arg \min_{\mathbf{Z}} G(\mathbf{Z}, \mathbf{Z}_{ab}^t) = \mathbf{Z}_{ab}^t \frac{(\mathbf{W}^T \mathbf{X}^T \mathbf{U})_{ab}}{(\mathbf{W}^T \mathbf{W} \mathbf{Z} \mathbf{U}^T \mathbf{U})_{ab}} \quad (\text{B.29})$$

As shown above, (B.25) is an auxiliary function and  $F_{ab}(\mathbf{Z})$  is non-unceasing under the optimization rule. □

## B.4 Convergence Proof of VNMF

**Theorem B.4.1.** *The objective function in 4.4 is non-increasing under the update rules (4.12) and Eq.(4.13). The objective function is invariant under these update rules if and only if  $\mathbf{U}$ , and  $\mathbf{V}$  are at a stationary point.*

**Lemma B.4.2.** *If there exists an auxiliary function  $G$  for  $F(v)$  with the properties  $G(v, v_0) \geq F(v)$  and  $G(v, v) = F(v)$ , then  $F$  is non-increasing under the update*

$$v^{(t+1)} = \arg \min_v G(v, v^t) \quad (\text{B.30})$$

*Proof.*  $F(v^{(t+1)}) \leq G(v^{(t+1)}, v^t) \leq G(v^t, v^t) = F(v^t)$  □

For any element  $\mathbf{V}_{ab}$  in  $\mathbf{V}$ , let  $F_{ab}$  denote the part of  $\mathcal{O}$  which is only relevant to  $\mathbf{V}_{ab}$ . Since the updating rules shown in (4.13) are element wise, showing the objective function is non-increasing is equal to show the  $F_{ab}$  is non-increasing.

we first compute the objective function relevant to the variable  $\mathbf{V}$  and its derivatives with respect to  $\mathbf{V}_{ab}$ :

$$F_{ab}(\mathbf{V}) = F_{ab}(\mathbf{V}_{ab}^t) + F'_{ab}(\mathbf{V} - \mathbf{V}_{ab}^t) + \frac{1}{2}F''_{ab}(\mathbf{V} - \mathbf{V}_{ab}^t)^2 \quad (\text{B.31})$$

$$F'_{ab} = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{V}}\right)_{ab} = (-2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U} - 4\lambda(\mathbf{N}\theta - \text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T))\mathbf{T}^T\mathbf{T}\mathbf{V})_{ab} \quad (\text{B.32})$$

$$F''_{ab} = 2(\mathbf{U}^T\mathbf{U})_{bb} - 4\lambda((\mathbf{N}\theta - \text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T))\mathbf{T}^T\mathbf{T})_{aa} - 4\lambda(\mathbf{T}^T\mathbf{T}\mathbf{V})_{aa}^2 \quad (\text{B.33})$$

**Lemma B.4.3.** *Let  $F_{ab}(\mathbf{V}_{ab})$  denote the part of  $F$  relevant to  $\mathbf{V}_{ab}$ . Then, the function*

$$G(\mathbf{V}, \mathbf{V}_{ab}^t) = F_{ab}(\mathbf{V}_{ab}^t) + F'_{ab}(\mathbf{V}_{ab}^t)(\mathbf{V} - \mathbf{V}_{ab}^t) + \frac{(\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\lambda(\text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T) - \mathbf{N}\theta) + (\mathbf{T}^+\mathbf{T}^+\mathbf{V} + \mathbf{T}^-\mathbf{T}^-\mathbf{V}))_{ab}}{\mathbf{V}_{ab}^t}(\mathbf{V} - \mathbf{V}_{ab}^t)^2 \quad (\text{B.34})$$

*is an auxiliary function for  $F_{ab}(\mathbf{V}_{ab})$ .*

**Proof.** It is obvious that  $G(\mathbf{V}_{ab}^t, \mathbf{V}_{ab}^t) = F_{ab}(\mathbf{V}_{ab}^t)$ . We only need to show that  $G(\mathbf{V}_{ab}, \mathbf{V}_{ab}^t) \geq F_{ab}(\mathbf{V}_{ab})$ . To do this, we expand  $F_{ab}(\mathbf{V})$  by Taylor series expansion

$$F_{ab}(\mathbf{V}) = F_{ab}(\mathbf{V}_{ab}^t) + F'_{ab}(\mathbf{V} - \mathbf{V}_{ab}^t) + \frac{1}{2}F''_{ab}(\mathbf{V} - \mathbf{V}_{ab}^t)^2 \quad (\text{B.35})$$

Therefore, it is only needed to prove that

$$\begin{aligned} & \frac{(\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\lambda(\text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T) - N\theta)(\mathbf{T}^+\mathbf{T}^+\mathbf{V} + \mathbf{T}^-\mathbf{T}^-\mathbf{V}))_{ab}}{\mathbf{V}_{ab}^t} \\ & \geq (\mathbf{U}^T\mathbf{U})_{bb} - 2\lambda((N\theta - \text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T))\mathbf{T}^T\mathbf{T})_{aa} \\ & \quad - 2\lambda(\mathbf{T}^T\mathbf{T}\mathbf{V})_{aa}^2 \end{aligned} \quad (\text{B.36})$$

We can expand  $\mathbf{V}\mathbf{U}^T\mathbf{U}$  as

$$(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} = \sum_{i=1}^k v_{ai}^t (\mathbf{U}^T\mathbf{U})_{ib} \geq v_{ab}^t (\mathbf{U}^T\mathbf{U})_{bb} \quad (\text{B.37})$$

For the rest terms, we have

$$\begin{aligned} (\mathbf{T}^+\mathbf{T}^+\mathbf{V}^t)_{ab} &= \sum_{c \neq a} (\mathbf{T}^+\mathbf{T}^+)_{ac} \mathbf{V}_{cb}^t + (\mathbf{T}^+\mathbf{T}^+)_{aa} \mathbf{V}_{ab}^t \\ &\geq (\mathbf{T}^T\mathbf{T})_{aa} \mathbf{V}_{ab}^t \end{aligned} \quad (\text{B.38})$$

As the  $\theta$  is set to a small value,  $\text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T)$  converge to  $N\theta$  from positive direction, we have

$$\text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T) - N\theta \geq 0 \quad (\text{B.39})$$

Thus, by multiplying Eq.B.38 with Eq.B.39 on both sides, we get

$$\begin{aligned} & 2\lambda((\text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T) - N\theta)\mathbf{T}^+\mathbf{T}^+\mathbf{V})_{ab} \\ & \geq -2\lambda((N\theta - \text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T))\mathbf{T}^T\mathbf{T})_{aa} v_{ab}^t \end{aligned} \quad (\text{B.40})$$

It is obvious that

$$2\lambda((\text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T) - N\theta)\mathbf{T}^-\mathbf{T}^-\mathbf{V})_{ab} \geq 0 \geq -2\lambda(\mathbf{T}^T\mathbf{T}\mathbf{V})_{aa}^2 \quad (\text{B.41})$$

Thus, the inequality of (B.36) holds because of the validation of (B.37), (B.40) and (B.41).  $\square$

**Proof of Theorem.B.4.1.** Replace  $G(v, v^t)$  in (B.30) with (B.34), we have the optimization rule

$$\begin{aligned} \mathbf{V}_{ab}^{(t+1)} &= \mathbf{V}_{ab}^t - \mathbf{V}_{ab}^t \frac{\mathbf{F}'_{ab}}{2(\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\lambda(\text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T) - N\theta)(\mathbf{T}^+\mathbf{T}^+\mathbf{V} + \mathbf{T}^-\mathbf{T}^-\mathbf{V}))_{ab}} \\ &= v_{ab} \frac{(\mathbf{X}^T\mathbf{U} + 4\lambda(\text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T) - N\theta)\mathbf{T}^+\mathbf{T}^-\mathbf{V})_{ab}}{(\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\lambda(\text{Tr}(\mathbf{T}\mathbf{V}\mathbf{V}^T\mathbf{T}^T) - N\theta)(\mathbf{T}^+\mathbf{T}^+\mathbf{V} + \mathbf{T}^-\mathbf{T}^-\mathbf{V}))_{ab}} \end{aligned} \quad (\text{B.42})$$

As proved above, (B.34) is an auxiliary function,  $\mathbf{F}_{ab}$  is non-increasing under this optimization rule.  $\square$



## B.5 Convergence Proof of Pairwise-NMF

**Theorem B.5.1.** *The objective function in (4.25) is non-increasing under the update rules (4.26), (4.27). The objective function is invariant under these update rules if and only if  $\mathbf{U}$ ,  $\mathbf{V}$  are at a stationary point.*

Here we prove the convergence of the derived update rules. Since the newly introduced terms depend only on  $\mathbf{V}$ , the update rule for  $\mathbf{U}$  remains the same as in the original NMF algorithm [LS01]. For the convergence proof of the proposed update rule for  $\mathbf{V}$ , we follow a similar procedure as in [LS01].

**Lemma B.5.2.** *If there exists an auxiliary function  $G$  for  $F(\mathbf{x})$  with the properties  $G(\mathbf{x}, \mathbf{x}') \geq F(\mathbf{x})$  and  $G(\mathbf{x}, \mathbf{x}) = F(\mathbf{x})$ , then  $F$  is non-increasing under the update*

$$\mathbf{x}^{t+1} = \arg \min_{\mathbf{x}} G(\mathbf{x}, \mathbf{x}'). \quad (\text{B.43})$$

**Proof.**  $F(\mathbf{x}^{t+1}) \leq G(\mathbf{x}^{t+1}, \mathbf{x}^t) \leq G(\mathbf{x}^t, \mathbf{x}^t) = F(\mathbf{x}^t)$  □

We first compute the objective function relevant to the variable  $\mathbf{V}$  and its derivatives with respect to  $\mathbf{V}_{ab}$ :

$$\begin{aligned} F(\mathbf{V}) &= -2\text{Tr}(\mathbf{X}\mathbf{V}\mathbf{U}^T) + \text{Tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) \\ &\quad + \lambda_1 \text{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) + \lambda_1 \exp[-\beta \text{Tr}(\mathbf{V}^T\mathbf{L}^{(f)}\mathbf{V})] \end{aligned} \quad (\text{B.44})$$

where its first derivative is given by

$$\begin{aligned} F'_{ab}(\mathbf{V}) &= (-2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} + \lambda_1 (2\mathbf{L}\mathbf{V})_{ab} \\ &\quad - \lambda_1 2\beta \exp[-\beta \text{Tr}(\mathbf{V}^T\mathbf{L}^{(f)}\mathbf{V})] (\mathbf{L}^{(f)}\mathbf{V})_{ab} \end{aligned} \quad (\text{B.45})$$

and finally its second derivative is

$$\begin{aligned} F''_{ab}(\mathbf{V}) &= 2(\mathbf{U}^T\mathbf{U})_{bb} + \lambda_1 2\mathbf{L}_{aa} \\ &\quad + \lambda_1 2 \left[ 2\beta^2 (\mathbf{L}^{(f)}\mathbf{V})_{ab}^2 - \beta \mathbf{L}_{aa}^{(f)} \right] \\ &\quad * \exp[-\beta \text{Tr}(\mathbf{V}^T\mathbf{L}^{(f)}\mathbf{V})] \end{aligned} \quad (\text{B.46})$$

**Lemma B.5.3.** *Let  $F_{ab}(\mathbf{V}_{ab})$  denote the part of  $F$  relevant to  $\mathbf{V}_{ab}$ . Then, the function*

$$\begin{aligned} G(\mathbf{V}, \mathbf{V}^t) &= F_{ab}(\mathbf{V}^t) + F'_{ab}(\mathbf{V}^t)(\mathbf{V} - \mathbf{V}^t)_{ab} \\ &\quad + \frac{1}{\mathbf{V}_{ab}^t} \{ (\mathbf{V}^t\mathbf{U}^T\mathbf{U})_{ab} + \lambda_1 (\mathbf{L}^+\mathbf{V}^t)_{ab} \\ &\quad + \lambda_1 \beta \exp[-\beta \text{Tr}((\mathbf{V}^t)^T\mathbf{L}^{(f)}\mathbf{V}^t)] \\ &\quad * (\mathbf{L}^{(f)-}\mathbf{V}^t)_{ab}^+ \} (\mathbf{V} - \mathbf{V}^t)_{ab}^2. \end{aligned} \quad (\text{B.47})$$

is an auxiliary function for  $F_{ab}(\mathbf{V}_{ab})$ .

We introduce the auxiliary function.

It is straightforward to check that  $G(\mathbf{V}^t, \mathbf{V}^t) = F(\mathbf{V}^t)$ . For the condition  $G(\mathbf{V}, \mathbf{V}^t) \geq F(\mathbf{V})$ , we compare the auxiliary function to the Taylor series expansion

$$\begin{aligned} F_{ab}(\mathbf{V}) &= F_{ab}(\mathbf{V}^t) + F'_{ab}(\mathbf{V}^t)(\mathbf{V} - \mathbf{V}^t)_{ab} \\ &\quad + \frac{1}{2}F''_{ab}(\mathbf{V} - \mathbf{V}^t)_{ab}^2 + O(\mathbf{V}_{ab}^3). \end{aligned} \quad (\text{B.48})$$

Comparing the second order terms of the auxiliary function with the second order terms of the Taylor series expansion, we get the condition:

$$\begin{aligned} &\frac{(\mathbf{V}^t \mathbf{U}^T \mathbf{U})_{ab}}{\mathbf{V}_{ab}^t} + \frac{\lambda_1 (\mathbf{L}^+ \mathbf{V}^t)_{ab}}{\mathbf{V}_{ab}^t} \\ &+ \frac{\lambda_1 \beta (\mathbf{L}^{(f)} - \mathbf{V}^t)_{ab} \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right]}{\mathbf{V}_{ab}^t} \\ &\geq (\mathbf{U}^T \mathbf{U})_{bb} + \lambda_1 \mathbf{L}_{aa} \\ &+ \lambda_1 \beta \left[ 2\beta (\mathbf{L}^{(f)} \mathbf{V}^t)_{ab}^2 - \mathbf{L}_{aa}^{(f)} \right] \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right]. \end{aligned} \quad (\text{B.49})$$

We now check the inequality for each term on the left side of the equation with its corresponding term on the right side. For the NMF-term, we have:

$$\begin{aligned} (\mathbf{V}^t \mathbf{U}^T \mathbf{U})_{ab} &= \sum_c \mathbf{V}_{ac}^t (\mathbf{U}^T \mathbf{U})_{cb} \\ &= \mathbf{V}_{ab}^t (\mathbf{U}^T \mathbf{U})_{bb} + \sum_{c \neq b} \mathbf{V}_{ac}^t (\mathbf{U}^T \mathbf{U})_{cb} \\ &\Rightarrow \frac{(\mathbf{V}^t \mathbf{U}^T \mathbf{U})_{ab}}{\mathbf{V}_{ab}^t} \geq (\mathbf{U}^T \mathbf{U})_{bb}. \end{aligned} \quad (\text{B.50})$$

For the similarity-term:

$$\begin{aligned} (\mathbf{L}^+ \mathbf{V}^t)_{ab} &\geq (\mathbf{L}^+ \mathbf{V}^t)_{ab} - (\mathbf{L}^- \mathbf{V}^t)_{ab} \\ &= (\mathbf{L} \mathbf{V}^t)_{ab} = \mathbf{L}_{aa} \mathbf{V}_{ab}^t + \sum_{c \neq b} \mathbf{L}_{ac} \mathbf{V}_{cb}^t \\ &\Rightarrow \lambda_1 \frac{(\mathbf{L}^+ \mathbf{V}^t)_{ab}}{\mathbf{V}_{ab}^t} \geq \lambda_1 \mathbf{L}_{aa}. \end{aligned} \quad (\text{B.51})$$

The inequality of the dissimilarity-term depends on the parameter  $\beta$ . For decreasing

$\beta$ , we have:

$$\begin{aligned}
 \frac{(\mathbf{L}^{(f)} - \mathbf{V}^t)_{ab}}{\mathbf{V}_{ab}^t} &\geq 2\beta (\mathbf{L}^{(f)} \mathbf{V}^t)_{ab}^2 - \mathbf{L}_{aa}^{(f)} \\
 \Rightarrow \lambda_1 \beta \frac{(\mathbf{L}^{(f)} - \mathbf{V}^t)_{ab} \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right]}{\mathbf{V}_{ab}^t} & \tag{B.52} \\
 &\geq \lambda_1 \beta \left[ 2\beta (\mathbf{L}^{(f)} \mathbf{V}^t)_{ab}^2 - \mathbf{L}_{aa}^{(f)} \right] * \exp \left[ -\beta \text{Tr} \left( (\mathbf{V}^t)^T \mathbf{L}^{(f)} \mathbf{V}^t \right) \right].
 \end{aligned}$$

Since the inequality holds for each term of the equation for the right choice of parameter  $\beta$ , it does also hold for the sum of all terms and (B.49) is true. For the higher order terms of the Taylor series expansion we note, that the derivatives disappear for the NMF- and similarity-term and that all derivatives of the dissimilarity-term are scaled by the factor  $\beta$ . Therefore, for decreasing  $\beta$  those also become negligible and the condition  $G(\mathbf{V}, \mathbf{V}^t) \geq F(\mathbf{V})$  is true.

***Proof of Theorem B.5.1.*** Inserting the auxiliary function (B.47) into (B.43) leads to the update rule (4.27). Thus, Theorem B.5.1 is true for (4.27). □



---

# Acronyms

ML	Machine Learning.
NMF	Non-negative Matrix Factorization.
SAR	Synthetic Aperture Radar.
DR	Dimensionality Reduction.
VDM	Visual Data Mining.
VR	Virtual Reality.
CAVE	Cave Automatic Virtual Environment.
IVR	Immersive Virtual Reality.
RF	Relevance Feedback.
AL	Active Learning.
QE	Query by Example.
GNMF	Graph Regularized Non-negative Matrix Factorization.
DGNMF	Dual Graph Regularized Non-negative Matrix Factorization.
MGNMF	Multiple Graph Regularized Non-negative Matrix Factorization.
DNMF	Discriminative Non-negative Matrix Factorization.
CNMF	Constrained Non-negative Matrix Factorization.
VNMF	Variance Regularized Non-negative Matrix Factorization.
LCA	Subspace Learning via Locally Constrained A-optimal Non-negative Projection.
CMNMF	Center-Map Non-negative Matrix Factorization.
MDS	Multidimensional Scaling.

SVM	Support Vector Machines.
SVD	Singular Value Decomposition.
PCA	Principal Component Analysis.
KLD	Kullback-Leibler Divergence.
KKT	Karush-Kuhn-Tucker.
AC	Accuracy.
nMI	normalized Mutual Information.
KM	Kuhn-Munkres.
PubFig	Public Figure Face Database.
OSR	Outdoor Scene Recognition.
EO	Earth Observation.
SDK	Software Development Kit.
MST	Minimum Spanning Tree.
LLE	Locally Linear Embedding.
LE	Laplacian Eigenmap.
SNE	Stochastic Neighbor Embedding.
MAED	Manifold Adaptive Experimental Design.
TED	Transductive Experimental Design.
OED	Optimum Experimental Design.
OVR	One-Versus-Rest.
BoW	Bag-of-Word.
DL	Dictionary Learning.
DDL	Discriminative Dictionary Learning.
LCMC	Local Continuity Meta-Criterion.
T&C	Trustworthiness and Continuity.
MRRE	Mean Relative Rank Error.
HCI	Human-Computer Interaction.
WLD	Weber Local Descriptor.
MQS	Membership Query Synthesis.
SBSS	Stream-Based Selective Sampling.
PBAL	Pool-Based Active Learning.
QBC	Query-By-Committee.
EMC	Expected Model Change.
RLS	Regularized Least Squares.
kNN	$k$ -Nearest Neighbors.
LLR	Locally Linear Reconstruction.
FCWL	First Certain Wrong Labeled.
TC	Trace-norm regularized Classifier.
RBF	Radial Basis Function.
RKHS	Reproducing Kernel Hilbert Space.
EM	Electromagnetic.
RCS	Radar Cross-Section.

---

## List of Symbols

<b>X</b>	Data matrix.
<b>U</b>	Matrix of bases.
<b>L</b>	The Laplacian of a matrix.
$\widetilde{\mathbf{W}}$	Normalized <b>W</b> whose rows sum to 1.
<b>V</b>	Matrix of coefficients which is considered as new features (new data representation).
$\mathbf{x}_i$	The (i)th row of the matrix <b>X</b> .
<b>N</b>	The number of samples.
<b>D</b>	The length of feature vectors.
<b>I</b>	The identity matrix.
<b>K</b>	The length of new feature vectors.
$N_l$	Number of labeled samples.
<b>S</b>	Number of classes.
<b>Q</b>	Contains the label of samples.
<b>P</b>	Percentage of labeled samples.
$\mathcal{L}$	Lagrangian.
<b>A</b>	A linear transformation matrix.
$\mathcal{A}$	A set of attributes.
$a_m$	An attribute.
$\mathcal{O}_m$	A set of ordered images pairs based on attribute $a_m$ .
$\mathcal{S}_m$	A set of unordered images pairs based on attribute $a_m$ .
<b>D</b>	A Dictionary matrix.
$H(\cdot)$	Entropy of a cluster.
<b>Tr</b>	Trace operator applied to a matrix.
$\alpha$	Regularizer parameter.
$\alpha_{ij}$	The distance between points $i$ and $j$ in a high-dimensional space.

$\beta_{ij}$	The distance between points $i$ and $j$ in a low-dimensional space.
$ \cdot $	The cardinality number of a set.
$\theta$	The parameter to control the variance value in the VNMF algorithm.
$\mathbf{V}_l$	New representation of labeled data.
$\Phi$	Matrix of Lagrange multipliers for the Matrix $\mathbf{U}$ .
$\phi_{ik}$	The (i,k) element of the matrix $\Phi$ .
$\Psi$	Matrix of Lagrange multipliers for the Matrix $\mathbf{V}$ .
$\psi_{jk}$	The (j,k) element of the matrix $\Psi$ .
$y(\mathbf{x})$	The true label of $\mathbf{x}$ .
$\hat{y}(\mathbf{x})$	The predicted label of $\mathbf{x}$ .
$p(\mathbf{x})$	A probability density function.
$E_T$	Expectation over training data.
$E_L$	Expectation over labeled data.
$E$	Expectation over the conditional density $P(y \mathbf{x})$ .
$V(y_i)$	The number of committee members.
$\theta^{(c)}$	A committee members.



---

## List of Figures

1.1	The diagram of the proposed visual data mining system. The contents of image repository are represented by feature vectors and fed into three processing blocks, namely interactive dimensionality reduction, visualization, and active learning. All these three blocks are connected with the CAVE. The learning algorithms incorporate the user's feedback from the cave in the learning process and send the results again into the CAVE (i.e., human in the loop). . . . .	4
2.1	Application of CNMF, and DNMF on two-parabolas data set with different degrees of labeling; each row corresponds to one method. Each column represents the result of the same labeling degree; (a,d) original samples; (b,e) results of 40% labeling; (c,f) results of 100% labeling. . . . .	14
2.2	(a) The Yale Faces data set; (b) The PIE Faces data set. . . . .	15
2.3	Clustering results for Yale Faces dataset. First row shows clustering accuracy for different percentages of labeling: (a) 30%; (b) 50%; (c) 70%. Second row shows normalized mutual information for different percentages of labeling: (d) 30%; (e) 50%; (f) 70%. . . . .	18
2.4	Clustering results for Handwritten Digits data set. First row shows clustering accuracy for different percentages of labeling: (a) 30%; (b) 50%; (c) 70%. Second row shows normalized mutual information for different percentages of labeling: (d) 30%; (e) 50%; (f) 70%. . . . .	19
2.5	Clustering results for PIE Faces dataset. First row shows clustering accuracy for different percentages of labeling: (a) 30%; (b) 50%; (c) 70%. Second row shows normalized mutual information for different percentages of labeling: (d) 30%; (e) 50%; (f) 70%. . . . .	20
2.6	Convergence of NMF (blue) and DNMF (red) on three datasets.(a) Yale Faces; (b) Handwritten Digits; (c) PIE Faces. . . . .	24

2.7	Parameter analysis of DNMF on three datasets: a) Yale faces; b) Handwritten digit; c) PIE faces. . . . .	25
2.8	Locality preserving of DNMF with different degrees of labeling applied on the data sets: (a) Yale Faces; (b) PIE Faces; (c) Handwritten Digit.	26
2.9	Binary description of images versus their relative descriptions [PG11a]	27
2.10	Example images from the PubFig and OSR data sets. . . . .	27
2.11	The list of attributes used for each data set, along with the binary and relative attribute annotations [PG11a] . . . . .	28
2.12	2D visualization of the data sets computed by the proposed method (DNMF); (a) the OSR data set; (b) the PubFig data set. Images with the same attribute are located close to each other. . . . .	29
2.13	Clustering results computed by PCA (cyan), NMF (black), DNMF (blue) and original data (red) evaluated by accuracy (AC) and normalized mutual information (nMI). (a) and (b) show the AC and nMI for the OSR dataset, respectively. (c) and (d) show the AC and nMI for the PubFig dataset, respectively. . . . .	30
2.14	Signal transformations of $\mathbf{x}_i$ and $\mathbf{x}_j$ as close as possible to $q_i$ and $q_j$ . .	32
2.15	Convergence of the objective function for the two data sets . . . . .	34
2.16	Evaluation of $\lambda_2$ for the two data sets (from left to right, Pubfig, OSR).	35
2.17	The clustering results obtained from the proposed method with and without relative attributes for the PubFig (first column) and OSR (second column) data sets. The first and second rows show the accuracy and the normalized mutual information (nMI), respectively. . . . .	35
2.18	Clustering results for PubFig (first column) and OSR (second column) data sets for increasing dictionary sizes. . . . .	37
2.19	Clustering results for PubFig (first column) and OSR (second column) data sets for increasing training data. . . . .	38
3.1	(a) A schematic of the CAVE. (b) Cave’s devices (Infrared camera, 3D glasses, and projectors) . . . . .	42
3.2	The physical diagram of immersive virtual reality. The visualization system consists of three layers with different responsibility. The first layer comprises motion capture (tracking) system and control capturing. A master PC in the middle layer for the synchronization, and finally four systems for rendering for each wall of the CAVE. All systems are connected via an Ethernet network. . . . .	44
3.3	(a) A pair of shutter glasses with markers attached for tracking the user. (b) An Xbox 360 controller attached with markers to control the scene. . . . .	45

3.4	(a) The process of selecting a group of data points using a semi-transparent 3D sphere; (b) the number of selected feature points is shown to the user. (c,d) Navigating and exploring images in the CAVE by zooming in and out. . . . .	46
3.5	Two samples of the visualization of neighborhood graphs (or trees) in the immersive 3D virtual environment. (a) the neighborhood graph of a real data; (b) the Minimum Spanning Tree of a synthetic data set.	47
3.6	An immersive visualization system provides the user with a visual representation of data. Here, high-dimensional features are extracted from a database of Earth Observation images and are fed into a dimensionality reduction technique to be visualized in an immersive 3D virtual environment. . . . .	48
3.7	Two samples of the visualization of features space of optical and SAR image data sets. (a) optical data set is represented by SIFT feature and MDS performs the dimensionality reduction; (b) the SAR data set is represented by Gabor features and Isomap is Dimensionality Reduction (DR) technique. . . . .	48
3.8	The workflow of the proposed approach. While data points are transferred from a high-dimensional space to low-dimensional one, the ranking matrices are built from the data points. These matrices are merged together to build up the co-ranking matrix that is used to define a joint probability distribution. Mutual information computed from this probability distribution is used to assess the quality of dimensionality reduction (here, communication channel) [BDR13]. . .	51
3.9	The quality of dimensionality reduction applied to the Merced and Corel data sets represented by mutual information and entropy of co-ranking matrix. A combination of three different features (color-histogram, SIFT, and WLD) and three different DR techniques (LE, SNE, LLE) yields 9 feature-DR methods indexed from 1-9; a) results of the Merced data set; b) results of the Corel data set; c) plotted result of method 8 from the Merced data set; d) plotted result of method 8 from the Corel data set. . . . .	52
3.10	Immersive visualization of image collections in the CAVE . . . . .	58
3.11	Visualizations of the Caltech data set (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF.	61
3.12	Caltech data set analysis of different regularization parameters (a) Structure (b) Entropy (c) Overview . . . . .	61
3.13	Caltech data set convergence plots (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF without regularization (e) Structure regularization (f) Comparison of clustering accuracy for different regularization terms . . . . .	62

3.14	Visualizations of the Corel dataset (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF.	63
3.15	Corel dataset analysis of different regularization parameters (a) Structure (b) Entropy (c) Overview . . . . .	63
3.16	Corel dataset convergence plots (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF without regularization (e) Structure regularization (f) Comparison of clustering accuracy for different regularization terms . . . . .	64
3.17	Visualization of the SAR dataset (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF.	65
3.18	SAR dataset analysis of different regularization parameters (a) Structure (b) Entropy (c) Overview . . . . .	65
3.19	SAR dataset convergence plots (a) Combination of regularization terms (b) Entropy regularization (c) Overview regularization (d) NMF without regularization (e) Structure regularization (f) Comparison of clustering accuracy for different regularization terms . . . . .	66
4.1	(a) the visualization of the clustering result in the CAVE. Images are positioned around their cluster centers based on their distances. A sample image of each cluster is used to depict the cluster center. (b) and (c) show user interactions on a desktop. A mis-clustered image is connected to a semantically cluster center by a green line. (d) a mis-clustered image (the image with red border) is connected (green line) to the cluster center of target cluster (with blue border) for SAR images. This interaction updated the semantic similarity matrix $W$ , which is used in our novel NMF algorithms. . . . .	72
4.2	A schematic view of the proposed divide-and-conquer approach to get a new representation of the data for clustering. Here, the training data is mixed with each part of test data and is fed into VNMF/CMNMF to get new representation $\mathbf{V}$ . The k-means algorithm is applied on each $\mathbf{V}$ separately and the results are mixed as the final results of clustering. . . . .	79
4.3	The objects in blue are the training data, and in green are test data. The square and circle indicates different classes. The dash line shows the similarity interaction. The blue dash interactions are done by the user while training the data. The green dash interactions are done by the system while applying the locality property . . . . .	79

4.4	Clustering results of proposed algorithms VNMF and CMNMF as a function of number of interactions represented by accuracy (first column) and normalized mutual information (second column) compared with k-means clustering algorithm, PCA and NMF. The first, second, and third rows show the results of Mean Variance, Image Intensity and WLD features, respectively. . . . .	81
4.5	Clustering results of proposed algorithms VNMF and CMNMF as a function of dimension of subspace represented by accuracy (first column) and normalized mutual information (second column) compared with k-means clustering algorithm, PCA and NMF. The first, second, and third rows show the results of Mean Variance, Image Intensity and WLD features, respectively. . . . .	82
4.6	The convergence speed for NMF, VNMF and CMNMF applied to the SAR data set represented by a) Mean-Variance; b) Image Intensity; and c) WLD features. . . . .	83
4.7	Computation times of divide-and-conquer compared to batch processing for all three datasets. The experiments were executed on a desktop PC with an Intel Core2Quad 2,8GHz CPU and 8GB of RAM. The divide-and-conquer approach is on average four times faster than batch processing. . . . .	84
4.8	3D visualization of the data sets using PCA and connecting similar and dissimilar images with green and red links. a) snapshot of interface from a far distance. b) snapshot of interface from a close distance. Here, two images that are close together but don't belong to the same class are linked with a red line. . . . .	86
4.9	Clustering results for Caltech and Corel data sets. a) Caltech; b) Corel; c) The convergence rate of algorithm applied to the Caltech data set; d) The convergence of algorithm applied to the Corel data set. . . . .	87
4.10	Sample snapshots of creating convex hulls (sets) around similar images in a Virtual Reality environment. a) a desktop display is used for visualization and interaction; b) the CAVE is used for creating sets. . . . .	88
4.11	Clustering results for the SAR data set represented by WLD and SIFT features. First and second row show the accuracy and normalized mutual information, respectively. The first and second columns are the results of WLD and SIFT features, respectively. . . . .	89
5.1	The active learning cycle (source [Set10]) . . . . .	93
5.2	Overview of active learning scenarios and sample selection strategies . . . . .	93
5.3	Visualization of support vectors for a synthetic dataset. Black crosses denote the support vectors (a) Support Vectors of SVM classifier (b) Support Vectors of the TC, approximated by hinge loss function. In the TC, more features are considered support vectors. . . . .	115

5.4	Example list of images presented by the algorithm for an optical dataset. The first row contains images representing each class. The sample images are arranged in the columns, which correspond to the predicted class, with decreasing margin [Bab+14b]. . . . .	119
5.5	Snapshot of the user interface for the FCWL algorithm on SAR data set after 40 iterations . . . . .	120
5.6	Classification results of three features representing the SAR data set. The first and second columns show the mean and the standard deviation of accuracy, respectively. The first, second, and third rows show the results of the Gabor, SIFT, and WLD features, respectively.	122
5.7	The left column shows the resulting matrix rank and trace-norm of the matrix $\mathbf{W}$ for different values $\lambda_1$ for different features. The right column shows the resulting singular values of $\mathbf{W}$ for different $\lambda_1$ . The first, second, and third rows are the results of the Gabor, SIFT, and WLD features, respectively. . . . .	124
5.8	Classification accuracy of different active learning algorithms on SAR dataset. The first and second columns show the mean and the standard deviation of accuracy, respectively. The first, second and third rows show the results of Gabor, SIFT, and WLD features, respectively. . .	126
5.9	Matrix rank and trace-norm of FCWL algorithm with TC on SAR data represented by: (a) Gabor; (b) SIFT; and (c) WLD features. . .	127
A.1	Some sample images of the SAR image data set. Each image corresponds to one category. . . . .	134
A.2	Some sample images of the Caltech10 data set. Each image corresponds to one category. . . . .	135
A.3	The UC Merced Land Use is a manually labeled data set containing 21 classes of land-use scenes. Each image represents one sample of each group. . . . .	135
A.4	The Corel images data set. Each image represents one sample of each category. . . . .	136
A.5	Some sample images of the CMU PIE faces data set. Each image corresponds to one category. . . . .	136
A.6	Some sample images of the ORL faces data set. Each image corresponds to one category. . . . .	137
A.7	Some sample images of the PIE faces data set. Each image corresponds to one category. . . . .	137
A.8	Some sample images of the Handwritten Digits data set. Each image corresponds to one category. . . . .	138

---

## List of Tables

2.1	Computational complexity for each iteration of NMF and DNMF . . .	13
2.2	Clustering Results for Yale Faces dataset and 50% labeling: AC (%)	21
2.3	Clustering Results for Handwritten Digits data set and 50% labeling: AC (%) . . . . .	22
2.4	Clustering Results for PIE Faces dataset and 50% labeling: AC (%)	23
2.5	Clustering results of different methods on the PubFig dataset . . . . .	29
2.6	Clustering results of different methods on the OSR dataset . . . . .	31
2.7	Accuracy and normalized mutual information for several dictionary learning algorithms applied to the data sets . . . . .	36
2.8	Runtime (in seconds) for several dictionary learning algorithms applied to the data sets. . . . .	36
4.1	The statistics of created convex hulls. The percentage of similar images (C1-C6) in each convex hull is presented. Additionally, the total number of images in each set is given in the second column. The seven created convex hulls contain different number of images. The images are coming from 6 different classes (C1-C6). . . . .	88
5.1	Computational complexity and actual training times of classifiers on SAR data set represented by BoW of SIFT feature descriptors. . . . .	116
5.2	Predicted samples images presented to the user during each iteration	118





---

## References

- [Rui+98] Y. Rui et al. “Relevance feedback: a power tool for interactive content-based image retrieval”. In: *Circuits and Systems for Video Technology, IEEE Transactions on* 8.5 (1998), pp. 644–655.
- [Set10] B. Settles. “Active Learning Literature Survey”. In: *University of Wisconsin, Madison* (2010).
- [Jin+06] P. Y. Jing et al. “Semantic Image Browser: Bridging Information Visualization with Automated Intelligent Image Analysis”. In: *Visual Analytics Science And Technology, 2006 IEEE Symposium On*. 2006, pp. 191–198.
- [TC10] C. Trevor F and M. A. Cox. *Multidimensional Scaling*. CRC Press, 2010.
- [GX11] M.D. Gupta and J. Xiao. “Non-negative Matrix Factorization As A Feature Selection Tool For Maximum Margin Classifiers”. In: *Computer Vision and Pattern Recognition (CVPR), IEEE International Conference on*. IEEE. 2011, pp. 2841–2848.
- [He+05] X. He et al. “Face Recognition Using Laplacianfaces”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.3 (2005), pp. 328–340.
- [Jol05] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.
- [MS07] A. Mnih and R. Salakhutdinov. “Probabilistic matrix factorization”. In: *Advances in neural information processing systems*. 2007, pp. 1257–1264.
- [DHS12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.
- [LS99] D. D. Lee and H. S. Seung. “Learning The Parts Of Objects By Non-negative Matrix Factorization.” In: *Nature* 401.6755 (Oct. 1999), pp. 788–791. ISSN: 0028-0836.

- [WZ13] Y.-X. Wang and Y. Zhang. “Nonnegative Matrix Factorization: A Comprehensive Review”. In: *Knowledge and Data Engineering, IEEE Transactions on* 25.6 (2013), pp. 1336–1353.
- [EF13] S. Essid and C Févotte. “Smooth nonnegative matrix factorization for unsupervised audiovisual document structuring”. In: *Multimedia, IEEE Transactions on* 15.2 (2013), pp. 415–425.
- [Liu+13] J. Liu et al. “Multi-view clustering via joint nonnegative matrix factorization”. In: *Proc. of SDM*. Vol. 13. SIAM. 2013, pp. 252–260.
- [XLG03] W. Xu, X. Liu, and Y. Gong. “Document clustering based on non-negative matrix factorization”. In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM. 2003, pp. 267–273.
- [GP13] N. Gillis and R. J. Plemmons. “Sparse nonnegative matrix underapproximation and its application to hyperspectral image analysis”. In: *Linear Algebra and its Applications* 438.10 (2013), pp. 3991–4007.
- [JQ09] S. Jia and Y. Qian. “Constrained Non-negative Matrix Factorization For Hyperspectral Unmixing”. In: *Geoscience and Remote Sensing, IEEE Transactions on* 47.1 (2009), pp. 161–173.
- [LCS13] C. Lin, Z. Cheng, and D. Shih. “Music Enhancement Using Nonnegative Matrix Factorization with Penalty Masking”. In: *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*. IEEE. 2013, pp. 125–129.
- [Vol+14] C. Vollmer et al. “Sparse coding of human motion trajectories with non-negative matrix factorization”. In: *Neurocomputing* 124 (2014), pp. 22–32.
- [Jia+14] B. Jiang et al. “A Sparse Nonnegative Matrix Factorization Technique For Graph Matching Problems”. In: *Pattern Recognition* 47.2 (2014), pp. 736–747.
- [Cai+11] D. Cai et al. “Graph Regularized Non-negative Matrix Factorization For Data Representation”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33.8 (2011), pp. 1548–1560. ISSN: 0162-8828.
- [SJW12] F. Shang, L. Jiao, and F. Wang. “Graph Dual Regularization Non-negative Matrix Factorization For Co-clustering”. In: *Pattern Recognition* 45.6 (2012), pp. 2237–2250.
- [WBG13] J. J.-Y. Wang, H. Bensmail, and X. Gao. “Multiple Graph Regularized Nonnegative Matrix Factorization”. In: *Pattern Recognition* 46.10 (2013), pp. 2840–2847.

- 
- [Hai+12] L. Haifeng et al. “Constrained Non-negative Matrix Factorization For Image Representation”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.7 (July 2012), pp. 1299–1311.
- [Li+13] P. Li et al. “Subspace learning via locally constrained A-optimal non-negative projection”. In: *Neurocomputing* 115 (2013), pp. 49–62.
- [LS01] D. D. Lee and H. S. Seung. “Algorithms For Non-negative Matrix Factorization”. In: *Advances in neural information processing systems*. 2001, pp. 556–562.
- [BV09] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge university press, 2009.
- [GV96] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996. ISBN: 0-8018-5414-8.
- [Cai+06] D. Cai et al. “Orthogonal Laplacianfaces For Face Recognition”. In: *IEEE Transactions on Image Processing* 15.11 (2006), pp. 3608–3614.
- [Kuh55] Harold W Kuhn. “The Hungarian method for the assignment problem”. In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.
- [PG11a] D. Parikh and K. Grauman. “Relative attributes”. In: *Computer Vision (ICCV), IEEE International Conference on*. IEEE. 2011, pp. 503–510.
- [OT01a] A. Oliva and A. Torralba. “Modeling the shape of the scene: A holistic representation of the spatial envelope”. In: *International journal of computer vision* 42.3 (2001), pp. 145–175.
- [AEB06] M. Aharon, M. Elad, and A. Bruckstein. “K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation”. In: *Signal Processing, IEEE Transactions on* 54.11 (2006), pp. 4311–4322.
- [EAH99] K. Engan, S. O. Aase, and J. Husoy. “Frame based signal compression using method of optimal directions (MOD)”. In: *Circuits and Systems (ISCAS), Proc. of the IEEE International Symposium on*. Vol. 4. IEEE. 1999, pp. 1–4.
- [RSS10] I. Ramirez, P. Sprechmann, and G. Sapiro. “Classification and clustering via dictionary learning with structured incoherence and shared features”. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE. 2010, pp. 3501–3508.
- [GTM14] S. Gao, I.-H. Tsang, and Y. Ma. “Learning category-specific dictionary and shared dictionary for fine-grained image categorization”. In: *Image Processing, IEEE Transactions on* 23.2 (2014), pp. 623–634.

- [ZL10] Q. Zhang and B. Li. “Discriminative K-SVD for dictionary learning in face recognition”. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE. 2010, pp. 2691–2698.
- [JLD11] Z. Jiang, Z. Lin, and L. S. Davis. “Learning a discriminative dictionary for sparse coding via label consistent K-SVD”. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE. 2011, pp. 1697–1704.
- [YZF11] M. Yang, D. Zhang, and X. Feng. “Fisher discrimination dictionary learning for sparse representation”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 543–550.
- [Cai+14] S. Cai et al. “Support Vector Guided Dictionary Learning”. In: *Computer Vision (ECCV), European Conference on*. Springer, 2014, pp. 624–639.
- [RZE08] R. Rubinstein, Mi. Zibulevsky, and M. Elad. “Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit”. In: *CS Technion 40.8 (2008)*, pp. 1–15.
- [Kum+09] N. Kumar et al. “Attribute and Simile Classifiers for Face Verification”. In: *Computer Vision (ICCV), IEEE International Conference on*. Oct. 2009.
- [Wri+09] J. Wright et al. “Robust face recognition via sparse representation”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.2 (2009), pp. 210–227.
- [Low04] D. G. Lowe. “Distinctive Image Features From Scale-invariant Keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.
- [SGS10] K.E.A. van de Sande, T. Gevers, and C.G.M. Snoek. “Evaluating Color Descriptors For Object And Scene Recognition”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (2010).
- [Fog+08] J. Fogarty et al. “Cueflik: Interactive Concept Learning In Image Search”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: ACM, 2008, pp. 29–38. ISBN: 978-1-60558-011-1.
- [Pan+11] Y. Pang et al. “Summarizing tourist destinations by mining user-generated travelogues and photos”. In: *Computer Vision and Image Understanding* 115.3 (2011), pp. 352–363.
- [Tal+09] J. Talbot et al. “Ensemblematrix: Interactive Visualization To Support Machine Learning With Multiple Classifiers”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: ACM, 2009, pp. 1283–1292. ISBN: 978-1-60558-246-7.

- 
- [MPH09] L. J. van der Maaten, E. O. Postma, and H. J. van den Herik. “Dimensionality Reduction: A Comparative Review”. In: (2009).
- [Pan+13] Y. Pang et al. “Ranking graph embedding for learning to rerank”. In: *IEEE Transactions on Neural Networks and Learning Systems* 24.8 (2013), pp. 1292–1303.
- [PWY14] Y. Pang, S. Wang, and Y. Yuan. “Learning Regularized LDA by Clustering”. In: *IEEE Transactions on Neural Networks and Learning Systems* 25.12 (2014), pp. 2191–1303.
- [Pom+14] F. Pompili et al. “Two algorithms for orthogonal nonnegative matrix factorization with application to clustering”. In: *Neurocomputing* 141 (2014), pp. 15–25.
- [VH08] L. Van der Maaten and G. Hinton. “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9.2579-2605 (2008), p. 85.
- [ZWY15] J.-S. Zhang, C.-P. Wang, and Y.-Q. Yang. “Learning Latent Features By Nonnegative Matrix Factorization Combining Similarity Judgments”. In: *Neurocomputing* (2015).
- [TP91] M. A Turk and A. P. Pentland. “Face Recognition Using Eigenfaces”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 1991, pp. 586–591.
- [RS00] S. T. Roweis and L. K. Saul. “Nonlinear Dimensionality Reduction By Locally Linear Embedding”. In: *Science* 290.5500 (2000), pp. 2323–2326.
- [BN03] M. Belkin and P. Niyogi. “Laplacian Eigenmaps For Dimensionality Reduction And Data Representation”. In: *Neural computation* 15.6 (2003), pp. 1373–1396.
- [HR02a] G. E. Hinton and S. T. Roweis. “Stochastic neighbor embedding”. In: *In Advances in Neural Information Processing Systems*. 2002, pp. 833–840.
- [TDL00] J. B. Tenenbaum, V. De Silva, and J. C. Langford. “A Global Geometric Framework For Nonlinear Dimensionality Reduction”. In: *Science* 290.5500 (2000), pp. 2319–2323.
- [Ber+00] M. Bernstein et al. *Graph approximations to geodesics on embedded manifolds*. Tech. rep. Technical report, Department of Psychology, Stanford University, 2000.
- [BS02] M. Balasubramanian and E. L. Schwartz. “The isomap algorithm and topological stability”. In: *Science* 295.5552 (2002), pp. 7–7.
- [CB09] L. Chen and A. Buja. “Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis”. In: *Journal of the American Statistical Association* 104.485 (2009), pp. 209–219.

- [VK06] J. Venna and S. Kaski. “Local Multidimensional Scaling”. In: *Neural Networks* 19.6 (2006), pp. 889–899.
- [LV09] J. A. Lee and M. Verleysen. “Quality assessment of dimensionality reduction: Rank-based criteria”. In: *Neurocomputing* 72.7 (2009), pp. 1431–1443.
- [LV07] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer, 2007.
- [BD13] R. Bahmanyar and M. Datcu. “Measuring the Semantic gap based on a Communication channel model”. In: *Image Processing, IEEE International Conference on*. 2013.
- [CJ10] M. Chen and H. Jaenicke. “An information-theoretic framework for visualization”. In: *Visualization and Computer Graphics, IEEE Transactions on* 16.6 (2010), pp. 1206–1215.
- [LV08] J. A. Lee and M. Verleysen. “Rank-based quality assessment of nonlinear dimensionality reduction”. In: *Artificial Neural Networks, Proceedings of 16th European Symposium on*. 2008, pp. 49–54.
- [Jie+08] C. Jie et al. “A Robust Descriptor Based On Weber’s Law”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2008.
- [Mog+04] B. Moghaddam et al. “Visualization and user-modeling for browsing personal photo libraries”. In: *International Journal of Computer Vision* 56.1-2 (2004), pp. 109–130.
- [NW08] G. P. Nguyen and M. Worring. “Interactive Access To Large Image Collections Using Similarity-based Visualization”. In: *Journal of Visual Languages & Computing* 19.2 (2008), pp. 203–224.
- [Wan+10] R. Wang et al. “Visualizing image collections using high-entropy layout distributions”. In: *IEEE Transactions on Multimedia* 12.8 (2010), pp. 803–813.
- [FP05] L. Fei-Fei and P. Perona. “A bayesian hierarchical model for learning natural scene categories”. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on*. Vol. 2. 2005, pp. 524–531.
- [Ren61] A. Renyi. “On Measures Of Entropy And Information”. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. Berkeley, Calif.: University of California Press, 1961, pp. 547–561.
- [Wis99] J. A. Wise. “The ecological approach to text visualization”. In: *Journal of the American Society for Information Science* 50.13 (1999), pp. 1224–1233.

- 
- [SGL08] J. Stasko, C. Görg, and Z. Liu. “Jigsaw: supporting investigative analysis through interactive visualization”. In: *Information visualization 7.2* (2008), pp. 118–132.
- [Jeo+09] D. H. Jeong et al. “iPCA: An Interactive System for PCA-based Visual Analytics”. In: vol. 28. 3. Wiley Online Library. 2009, pp. 767–774.
- [Cho+13] J. Choo et al. “An Interactive Visual Testbed System For Dimension Reduction And Clustering Of Large-scale High-dimensional Data”. In: *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics. 2013, pp. 865402–865402.
- [Azz+05] H. Azzag et al. “Vrminer: A Tool For Multimedia Database Mining With Virtual Reality”. In: *Processing and Managing Complex Data for Decision Support* (2005), pp. 318–339.
- [NH01] M. Nakazato and T. S. Huang. “3d Mars: Immersive Virtual Reality For Content-based Image Retrieval.” In: *ICME*. 2001.
- [Hol12] A. Holzinger. “On knowledge discovery and interactive intelligent visualization of biomedical data-challenges in human-computer interaction & biomedical informatics”. In: *9th International Joint Conference on e-Business and Telecommunications (ICETE 2012)*, pp. IS9–IS20. 2012.
- [WXH11] B. W. Wong, K. Xu, and A. Holzinger. “Interactive visualization for information analysis in medical diagnosis”. In: *Information Quality in e-Health*. Springer, 2011, pp. 109–120.
- [CDD13] S. Cui, C. O. Dumitru, and M. Datcu. “Ratio-detector-based feature extraction for very high resolution SAR image patch indexing”. In: *Geoscience and Remote Sensing Letters, IEEE* 10.5 (2013), pp. 1175–1179.
- [MC13] V. Mayer-Schönberger and K. Cukier. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.
- [Pat14] L. B. Patra. “A Novel Som-svm Based Active Learning Technique For Image Classification”. In: *Geoscience and Remote Sensing*, 52 (2014), pp. 6899–6910.
- [Per+14] C. Persello et al. “Cost-sensitive Active Learning With Lookahead: Optimizing Field Surveys For Remote Sensing Data Classification”. In: *Geoscience and Remote Sensing, IEEE Transactions on* 52.10 (2014), pp. 6652–6664.
- [Set12] B. Settles. “Active Learning”. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6.1 (2012), pp. 1–114.

- [Tui+11] D. Tuia et al. “A Survey Of Active Learning Algorithms For Supervised Remote Sensing Image Classification”. In: *Selected Topics in Signal Processing, IEEE Journal of* 5.3 (2011), pp. 606–617.
- [WH11] M. Wang and X.-S. Hua. “Active Learning In Multimedia Annotation And Retrieval: A Survey”. In: *ACM Trans. Intell. Syst. Technol.* 2.2 (Feb. 2011). ISSN: 2157-6904.
- [CZJ96] D. A. Cohn, G. Zoubin, and M. I. Jordan. “Active Learning With Statistical Models”. In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 129–145.
- [TC01] S. Tong and E. Chang. “Support Vector Machine Active Learning for Image Retrieval”. In: *Proceedings of the Ninth ACM International Conference on Multimedia*. ACM, 2001, pp. 107–118.
- [JPP09] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. “Multi-class Active Learning For Image Classification.” In: *CVPR*. IEEE, 2009.
- [SOS92] H. S. Seung, M. Opper, and H. Sompolinsky. “Query by Committee”. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. ACM, 1992.
- [SCS08] B. Settles, M. Craven, and R. Soumya. “Multiple-instance Active Learning”. In: *In Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2008, pp. 1289–1296.
- [DRH06] C. K. Dagli, S. Rajaram, and T. Huang. “Leveraging Active Learning For Relevance Feedback Using An Information Theoretic Diversity Measure.” In: *CIVR*. Springer, 2006.
- [Zha+09a] X. Zhang et al. “Multi-view Multi-label Active Learning For Image Classification.” In: *ICME*. IEEE, 2009.
- [NS04] H. T. Nguyen and A. Smeulders. “Active Learning Using Pre-clustering”. In: *Proceedings of the Twenty-first International Conference on Machine Learning*. ICML '04. ACM, 2004.
- [Qi+06] G.-J. Qi et al. “Video Annotation by Active Learning and Cluster Tuning”. In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*. 2006.
- [ADT07] A. C. Atkinson, A. N. Donev, and R. D. Tobias. *Optimum Experimental Designs, With Sas*. Oxford Statistical Science Series. New York: Oxford University Press, 2007. ISBN: 9780199296590.
- [CS00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. New York, NY, USA: Cambridge University Press, 2000. ISBN: 0-521-78019-5.



- 
- [EPP00] T. Evgeniou, M. Pontil, and T. Poggio. “Regularization Networks and Support Vector Machines”. In: *Advances in Computational Mathematics* (2000).
- [Alt92] N. S. Altman. “An Introduction To Kernel And Nearest-neighbor Non-parametric Regression”. In: *The American Statistician* 46.3 (1992), pp. 175–185.
- [YBT06] K. Yu, J. Bi, and V. Tresp. “Active Learning Via Transductive Experimental Design”. In: *Machine Learning (ICML), the 23rd International Conference on*. ICML ’06. Pittsburgh, Pennsylvania: ACM, 2006, pp. 1081–1088. ISBN: 1-59593-383-2.
- [Yu+08] K. Yu et al. “Non-greedy Active Learning For Text Categorization Using Convex Transductive Experimental Design”. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’08. Singapore, Singapore: ACM, 2008, pp. 635–642. ISBN: 978-1-60558-164-4.
- [Zha+11] L. Zhang et al. “Active Learning Based On Locally Linear Reconstruction”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33.10 (2011), pp. 2026–2038.
- [CH12] D. Cai and X. He. “Manifold Adaptive Experimental Design For Text Categorization”. In: *Knowledge and Data Engineering, IEEE Transactions on* 24.4 (Apr. 2012), pp. 707–719. ISSN: 1041-4347.
- [VP05] S. Vikas and N. Partha. “Beyond The Point Cloud: From Transductive To Semi-supervised Learning”. In: *In ICML*. 2005, pp. 824–831.
- [Har+12] Z. Harchaoui et al. “Large-scale Image Classification With Trace-norm Regularization”. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. June 2012, pp. 3386–3393.
- [RK04] R. Rifkin and A. Klautau. “In Defense of One-Vs-All Classification”. In: *J. Mach. Learn. Res.* 5 (Dec. 2004), pp. 101–141. ISSN: 1532-4435.
- [Vap82] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1982. ISBN: 0387907335.
- [CL11] C. Chang and C. Lin. “Libsvm: A Library For Support Vector Machines”. In: *ACM Trans. Intell. Syst. Technol.* 2.3 (May 2011).
- [Che05] K. Chen. *Matrix Preconditioning Techniques And Applications*. Oxford Statistical Science Series. Cambridge University Press, 2005.
- [LW02] C. Liu and H. Wechsler. “Gabor Feature Based Classification Using The Enhanced Fisher Linear Discriminant Model For Face Recognition”. In: *Image processing, IEEE Transactions on* 11.4 (2002), pp. 467–476.

- [CL13] Mark T Crockett and David Long. “An Introduction to Synthetic Aperture Radar: a High-Resolution Alternative to Optical Imaging”. In: (2013).
- [DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum Likelihood From Incomplete Data Via The Em Algorithm”. In: *Journal of the Royal Statistical Society Series B* 39.1 (1977), pp. 1–38.
- [Aka+13] Z. Akata et al. “Label-embedding for attribute-based classification”. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE. 2013, pp. 819–826.
- [ATH03] S. Andrews, I. Tsochantaridis, and T. Hofmann. “Support Vector Machines for Multiple-Instance Learning”. In: *Advances in Neural Information Processing Systems 15 (NIPS)*. Ed. by S. Thrun S. Becker and K. Obermayer. Cambridge, MA, USA: MIT Press, 2003, pp. 561–568.
- [AC75] M. Argyle and M. Cook. *Gaze and Mutual Gaze*. Cambridge, England: Cambridge University Press, 1975.
- [Ash+07] A. Ashraf et al. “The Painful Face: Pain Expression Recognition Using Active Appearance Models”. In: *Proceedings of the 9th International Conference on Multimodal Interfaces (ICMI), Special Session on Multimodal Analysis of Human Spontaneous Behaviour*. ACM SIGCHI. Nagoya, Japan, 2007, pp. 9–14.
- [Böh+03] M. Böhlen et al. “3d Visual Data Mining; Goals And Experiences”. In: *Computational statistics & data analysis* 43.4 (2003), pp. 445–469.
- [Bao+09] L. Bao et al. “Locally Non-negative Linear Structure Learning For Interactive Image Retrieval”. In: *Proceedings of the 17th ACM international conference on Multimedia*. ACM. 2009, pp. 557–560.
- [Bat+05] A. Batliner et al. “Private Emotions Vs. Social Interaction - Towards New Dimensions In Research On Emotion”. In: *Proceedings of the International Workshop on Adapting the Interaction Style to Affective Factors, at the 10th International Conference on User Modelling*. Edinburgh, Scotland, 2005.
- [BPP96] A. Berger, V. Pietra, and S. Pietra. “A Maximum Entropy Approach To Natural Language Processing”. In: *Computational linguistics* 22.1 (1996), pp. 39–71.
- [Ble+03] D. M. Blei et al. “Latent Dirichlet Allocation”. In: *Journal of Machine Learning Research* 3 (2003).
- [BBG13] S. Borgwardt, A. Brieden, and P. Gritzmann. “A Balanced K-means Algorithm For Weighted Point Sets”. In: *arXiv:1308.4004v1* (2013).

- 
- [Bru11] G. Brumfiel. “High-energy Physics: Down The Petabyte Highway”. In: *Nature* (2011).
- [BPG10] P. Bruneau, F. Picarougne, and M. Gelgon. “Interactive Unsupervised Classification And Visualization For Browsing An Image Collection”. In: *Pattern Recognition* 43.2 (2010), pp. 485–493.
- [Che06] L. Chen. “Local Multidimensional Scaling For Nonlinear Dimensionality Reduction, Graph Layout, And Proximity Analysis”. PhD thesis. University of Pennsylvania, 2006.
- [Cic+09] A. Cichocki et al. *Non-negative Matrix And Tensor Factorizations: Applications To Exploratory Multi-way Data Analysis And Blind Source Separation*. John Wiley & Sons, 2009.
- [Cow+01] R. Cowie et al. “Emotion Recognition In Human-computer Interaction”. In: *IEEE Signal Processing magazine* 18.1 (January 2001), pp. 32–80.
- [cri14] crisp. *Image for SAR Imaging Geometry*, retrieved 23.08.2014. 2014. URL: <http://www.crisp.nus.edu.sg/~research/tutorial/sargm.gif>.
- [Cuk10] Kenneth Cukier. “Data, data everywhere”. In: *The Economist* (2010).
- [CM91] J.C. Curlander and R.N. McDonough. *Synthetic Aperture Radar: Systems and Signal Processing*. Wiley Series in Remote Sensing and Image Processing. Wiley, 1991.
- [DB14] B. Demir and L. Bruzzone. “A multiple criteria active learning method for support vector regression”. In: *Pattern Recognition* 47.7 (2014), pp. 2558–2567.
- [DMB14] B. Demir, L. Minello, and L. Bruzzone. “An Effective Strategy To Reduce The Labeling Cost In The Definition Of Training Sets By Active Learning”. In: *Geoscience and Remote Sensing Letters, IEEE* 11.1 (2014), pp. 79–83.
- [Ela10] M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.
- [Fér+05] N. Férey et al. “Visual data mining of genomic databases by immersive graph-based exploration”. In: *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*. ACM. 2005, pp. 143–146.
- [Far+] A. Farhadi et al. “Describing Objects By Their Attributes”. In: *Computer Vision and Pattern Recognition (CVPR). IEEE Conference on*, pp. 1778–1785.

- [Gat+05] D. Gatica-Perez et al. “Detecting Group Interest-level In Meetings”. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Philadelphia, PA, USA, Mar. 2005.
- [Gen+08] B. Geng et al. “Unbiased active learning for image retrieval”. In: *Multimedia and Expo, 2008 IEEE International Conference on*. IEEE. 2008, pp. 1325–1328.
- [Ham+04] J. Ham et al. “A Kernel View Of The Dimensionality Reduction Of Manifolds”. In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 47.
- [HR02b] G. Hinton and S. Roweis. “Stochastic neighbor embedding”. In: *Advances in neural information processing systems* 15 (2002), pp. 833–840.
- [Joa02] T. Joachims. “Optimizing search engines using clickthrough data”. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2002, pp. 133–142.
- [JJ09] S. Johansson and J. Johansson. “Interactive Dimensionality Reduction Through User-defined Combinations Of Quality Metrics”. In: *Visualization and Computer Graphics, IEEE Transactions on* 15.6 (2009), pp. 993–1000.
- [KPI04] A. Kapoor, R.W. Picard, and Y. Ivanov. “Probabilistic Combination of Multiple Modalities to Detect Interest”. In: *Proceedings of the 19th International Workshop on Pattern Recognition, (ICPR)*. Vol. 3. Cambridge, United Kingdom, 2004, pp. 969–972.
- [KP11] S. Kaski and J. Peltonen. “Dimensionality Reduction for Data Visualization”. In: *Signal Processing Magazine* 28.2 (2011), pp. 100–104.
- [KE03] L. Kennedy and D. Ellis. “Pitch-based emphasis detection for characterization of meeting recordings”. In: *Proceedings of the International Workshop on Automatic Speech Recognition and Understanding (ASRU)*. St Thomas, VI, USA, Dec. 2003.
- [KG13] A. Kovashka and K. Grauman. “Attribute pivots for guiding relevance feedback in image search”. In: *Computer Vision (ICCV), IEEE International Conference on*. IEEE. 2013, pp. 297–304.
- [KPG12] A. Kovashka, D. Parikh, and K. Grauman. “Whittlesearch: Image search with relative attribute feedback”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 2973–2980.
- [KVG] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. “Actively Selecting Annotations Among Objects And Attributes”. In: *Computer Vision (ICCV), IEEE International Conference on*, pp. 1403–1410.

- 
- [Kum+] N. Kumar et al. “Attribute And Simile Classifiers For Face Verification”. In: *Computer Vision (ICCV), IEEE International Conference on*, pp. 365–372.
- [Lee+06] H. Lee et al. “Efficient sparse coding algorithms”. In: *Advances in neural information processing systems*. 2006, pp. 801–808.
- [LG94] D. D. Lewis and W. A. Gale. “A Sequential Algorithm For Training Text Classifiers”. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '94. Dublin, Ireland: Springer-Verlag New York, Inc., 1994, pp. 3–12. ISBN: 0-387-19889-X.
- [LLC14] C. Li, H. Liu, and D. Cai. “Active Learning On Manifolds.” In: *Neuro-computing* 123 (2014).
- [LC07] J. Li and W. Chen. “Clustering synthetic aperture radar (SAR) imagery using an automatic approach”. In: *Canadian Journal of Remote Sensing* 33.4 (2007), pp. 303–311.
- [LN89] D. C. Liu and J. Nocedal. “On the Limited Memory BFGS Method for Large Scale Optimization”. In: *Math. Program.* (1989).
- [LS96] N. K. Logothetis and D. L. Sheinberg. “Visual Object Recognition.” In: *Annual Review of Neuroscience* 19 (1996), pp. 577–621.
- [LP86] L. Lovász and M. Plummer. *Matching Theory*. Akadémiai Kiadó, Budapest, 1986.
- [Low99] D.G. Lowe. “Object Recognition From Local Scale-invariant Features”. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. 1999.
- [MP07] L. Maat and M. Pantic. “Gaze-x: Adaptive, Affective, Multimodal Interface For Single-user Office Scenarios”. In: *Artificial Intelligence for Human Computing*. Springer, 2007, pp. 251–271.
- [Mac03] D. J. MacKay. *Information theory, inference, and learning algorithms*. Vol. 7. Citeseer, 2003.
- [Ma67] J. MacQueen and et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.
- [Mac67] J. B. MacQueen. “Some Methods for Classification and Analysis of MultiVariate Observations”. In: *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. Ed. by L. M. Le Cam and J. Neyman. Vol. 1. University of California Press, 1967, pp. 281–297.

- [Mai+09] J. Mairal et al. “Online dictionary learning for sparse coding”. In: *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*. ACM. 2009, pp. 689–696.
- [Mal99] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.
- [MM96] B.S. Manjunath and W.Y. Ma. “Texture features for browsing and retrieval of image data”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18.8 (1996), pp. 837–842. ISSN: 0162-8828.
- [MP03] S. Mota and Picard. “Automated Posture Analysis For Detecting Learner’s Interest Level”. In: *Proceedings of the International Workshop on Computer Vision and Pattern Recognition (CVPR) for HCI*. Madison, WI, USA, June 2003.
- [NGM01] H. R. Nagel, E. Granum, and P. Musaeus. “Methods For Visual Mining Of Data In Virtual Reality”. In: *Proceedings of the International Workshop on Visual Data Mining*. 2001, pp. 13–27.
- [NL86] J.-L. Nespoulous and A. Lecours. “Gestures: Nature and Function”. In: *The Biological Foundations Of Gestures: Motor And Semiotic Aspects*. Ed. by J.-L. Nespoulous, P. Perron, and A.R. Lecours. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1986, pp. 49–62.
- [Nie+02] R. Nieschulz et al. “Aspects Of Efficient Usability Engineerings”. In: *it+ti journal, "Usability Engineering"* 44.1 (2002), pp. 23–30.
- [OT01b] A. Oliva and A. Torralba. “Modeling The Shape Of The Scene: A Holistic Representation Of The Spatial Envelope”. In: *International journal of computer vision* 42.3 (2001), pp. 145–175.
- [PGC06] N. Panda, K.-S. Goh, and E. Y. Chang. “Active learning in very large databases”. In: *Multimedia Tools and Applications* 31.3 (2006), pp. 249–267.
- [PLY10] Y. Pang, X. Li, and Y. Yuan. “Robust tensor analysis with L1-norm”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 20.2 (2010), pp. 172–178.
- [PR03] M. Pantic and L. Rothkrantz. “Toward an Affect-Sensitive Multimodal Human-Computer Interaction”. In: *Proceedings of the IEEE 91* (September 2003), pp. 1370–1390.
- [PG11b] D. Parikh and K. Grauman. “Relative attributes”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 503–510.
- [PB12a] S. Patra and L. Bruzzone. “A Batch-mode Active Learning Technique Based On Multiple Uncertainty For Svm Classifier”. In: *Geoscience and Remote Sensing Letters, IEEE* 9.3 (2012), pp. 497–501.

- 
- [PB12b] S. Patra and L. Bruzzone. “A Cluster-assumption Based Batch Mode Active Learning Technique”. In: *Pattern Recognition Letters* 33.9 (2012), pp. 1042–1048.
- [PM05] A. Pentland and A. Madan. “Perception of social interest”. In: *Proceedings of the 10th International Conference on Computer Vision, Workshop on Modeling People and Human Interaction (ICCV-PHI)*. Beijing, China, Oct. 2005.
- [Phi+08] J. Philbin et al. “Lost in quantization: Improving particular object retrieval in large scale image databases”. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE. 2008, pp. 1–8.
- [QBZ05] P. Qvarfordt, D. Beymer, and S. X. Zhai. “RealTourist - A Study of Augmenting Human-Human and Human-Computer Dialogue with Eye-Gaze Overlay”. In: *Human-Computer Interaction - INTERACT 2005*. Vol. LNCS 3585. Springer Berlin / Heidelberg, 2005, pp. 767–780.
- [RW84] R. A. Redner and H. F. Walker. “Mixture Densities, Maximum Likelihood And The Em Algorithm”. In: *SIAM review* 26.2 (1984), pp. 195–239.
- [Roc71] J. J. Rocchio. “Relevance feedback in information retrieval”. In: (1971).
- [Sah+08] H. Sahbi et al. “Manifold Learning Using Robust Graph Laplacian For Interactive Image Search”. In: *Computer Vision and Pattern Recognition (CVPR). IEEE Conference on*. IEEE. 2008, pp. 1–8.
- [SSM97] B. Schölkopf, A. Smola, and K.-R. Müller. “Kernel principal component analysis”. In: *Artificial Neural Networks*. Springer, 1997, pp. 583–588.
- [Sch+09] M. Schröder et al. “A Demonstration Of Audiovisual Sensitive Artificial Listeners”. In: *Proceedings of the 4th International HUMAINE Association Conference on Affective Computing and Intelligent Interaction (ACII)*. Amsterdam, The Netherlands, 2009, pp. 263–264.
- [Sha48] C. E. Shannon. “A mathematical theory of communication”. In: *Bell System Technical Journal* 27 (1 (1948)), pp. 379–423.
- [Shr05] E. Shriberg. “Spontaneous speech: How people really talk and why engineers should care”. In: *Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech)*. Lisbon, Portugal, Sept. 2005.
- [SBM08] S. Simoff, M. Böhlen, and A. Mazeika. *Visual data mining: theory, techniques and tools for visual analytics*. Vol. 4404. Springer, 2008.

- [Ski88] J. Skilling. “The Axioms of Maximum Entropy”. In: *Maximum-Entropy and Bayesian Methods in Science and Engineering* ((1988)). Ed. by Erickson and Smith.
- [Sme+00] A.W.M. Smeulders et al. “Content-based Image Retrieval At The End Of The Early Years”. In: *Pattern Analysis and Machine Intelligence*, 22.12 (2000), pp. 1349–1380.
- [Spe12] N. Spencer. “How Much Data is Created Every Minute?” In: *Visual News* (2012).
- [SYW02] R. Stiefelhagen, J. Yang, and A. Waibel. “Modeling Focus Of Attention For Meeting Indexing Based On Multiple Cues”. In: *IEEE Transactions on Neural Networks* 13.4 (July 2002), pp. 928–938.
- [Tho93] K. Thorisson. “Dialogue Control In Social Interface Agents”. In: *Proceedings of the Conference Companion on Human factors in Computing Systems (CHI)*. Amsterdam, The Netherlands: ACM, 1993, pp. 139–140.
- [TKB92] K. Thorisson, D. Koons, and R. Bolt. “Multi-modal Natural Dialogue”. In: *Proceedings of the Special Interest Group on Computer-Human Interaction (SIGCHI) Conference on Human factors in Computing Systems (CHI)*. Monterey, CA, United States: ACM, 1992, pp. 653–654.
- [VG09] S. Vijayanarasimhan and K. Grauman. “What’s it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations”. In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE. 2009, pp. 2262–2269.
- [WAG12] J.-Y. Wang, I. Almasri, and X. Gao. “Adaptive Graph Regularized Non-negative Matrix Factorization Via Feature Selection”. In: *Pattern Recognition (ICPR), 21st International Conference on*. IEEE. 2012, pp. 963–966.
- [Wan+07] M. Wang et al. “Interactive Video Annotation By Multi-concept Multimodality Active Learning”. In: *International Journal of Semantic Computing* 1.04 (2007), pp. 459–477.
- [WSS04] K. Q. Weinberger, F. Sha, and L. K. Saul. “Learning A Kernel Matrix For Nonlinear Dimensionality Reduction”. In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 106.
- [WLM11] D. Wijayasekara, O. Linda, and M. Manic. “Cave-som: Immersive Visual Data Mining Using 3d Self-organizing Maps”. In: *Neural Networks (IJCNN), The International Joint Conference on*. IEEE. 2011, pp. 2471–2478.



- 
- [YYH03] R. Yan, L. Yang, and A. Hauptmann. “Automatically labeling video data using multi-class active learning”. In: *Computer Vision (ICCV), IEEE International Conference on*. IEEE. 2003, pp. 516–523.
- [YN10] Y. Yang and S. Newsam. “Bag-of-visual-words And Spatial Extensions For Land-use Classification”. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '10. ACM, 2010, pp. 270–279.
- [Zal12] D. Zall. “Visual Data Mining-An Approach to Hybrid 3D Visualization”. In: (2012).
- [Zha+09b] L. Zhang et al. “Convex Experimental Design Using Manifold Structure For Image Retrieval”. In: *Proceedings of the 17th ACM international conference on Multimedia*. ACM. 2009, pp. 45–54.
- [ZZ03] Z. Zhou and M. Zhang. “Ensembles Of Multi-instance Learners”. In: *Proceedings of the 14th European Conference on Machine Learning*. Springer, 2003, pp. 492–502.
- [ZLG03] X. Zhu, J. Lafferty, and Z. Ghahramani. “Combining Active Learning And Semi-supervised Learning Using Gaussian Fields And Harmonic Functions”. In: *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*. 2003, pp. 58–65.



---

## Publications by Author

- [Bab+13a] M. Babae et al. “Immersive Visual Information Mining For Exploring The Content Of EO Archives”. In: *International Conference on Big Data from Space*. 2013.
- [Bab+13b] M. Babae et al. “Immersive Visual Information Mining For Exploring The Content Of EO Archives”. In: *European Space Agency Living Planet Symposium*. 2013.
- [Bab+13c] M. Babae et al. “Immersive Visual Information Mining For Exploring The Content Of Terrasar-x Archives”. In: *5. TerraSAR-X Science Team Meeting*. 2013.
- [Bab+ara] M. Babae et al. “Discriminative Non-negative Matrix Factorization For Dimensionality Reduction”. In: *Elsevier Neurocomputing* (to appear).
- [BRD13a] M. Babae, G. Rigoll, and M. Datcu. “Immersive Interactive Information Mining With Application To Earth Observation Data Retrieval”. In: *Availability, Reliability, and Security in Information Systems and HCI*. Springer, 2013, pp. 376–386.
- [Bab+arb] M. Babae et al. “Immersive Visualization Of Visual Data Using Non-negative Matrix Factorization”. In: *Elsevier Neurocomputing* (to appear).
- [Bab+14a] M. Babae et al. “Immersive Visualization Of SAR Images Using Non-negative Matrix Factorization”. In: *Big Data from Space, International Conference on*. 2014.
- [Bab+15a] M. Babae et al. “Active Learning Using A Low-rank Classifier”. In: *Iranian Conference on Electrical Engineering, ICEE’15*. IEEE, May 2015.
- [Bab+15b] M. Babae et al. “Visualization-based Active Learning For The Annotation Of Sar Images”. In: *Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS), IEEE journal of* (2015).

- [Bab+14b] M. Babae et al. “Interactive Visualization Based Active Learning”. In: *Human-Machine Communication for Visual Recognition and Search (HMCV) Workshop, held in conjunction with the European Conference on Computer Vision (ECCV)*. 2014.
- [Bab+14c] M. Babae et al. “Farness preserving Non-negative matrix factorization”. In: *Image Processing (ICIP), IEEE International Conference on*. IEEE. 2014, pp. 3023–3027.
- [Bab+15c] M. Babae et al. “Attribute Constrained Subspace Learning”. In: *Image Processing (ICIP), IEEE International Conference on*. 2015.
- [Bab+14d] M. Babae et al. “Dimensionality Reduction Using Relative Attributes”. In: *Third International Workshop in Parts and Attributes, held in conjunction with the European Conference on Computer Vision (ECCV)*. 2014.
- [BDR13] M. Babae, M. Datcu, and G. Rigoll. “Assessment Of Dimensionality Reduction Based On Communication Channel Model; Application To Immersive Information Visualization”. In: *Big Data, IEEE International Conference on*. 2013, pp. 1–6.
- [Bab+15d] M. Babae et al. “Interactive Feature Learning from SAR Image Patches”. In: *International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2015.
- [Bab+14e] M. Babae et al. “Immersive Visual Analytics Of Earth Observation Data”. In: *Big Data from Space, International Conference on*. 2014.
- [Bab+14f] M. Babae et al. “Interactive Clustering For SAR Image Understanding”. In: *IEEE EUSAR*. 2014.
- [Bab+14g] M. Babae et al. “Locally Linear Salient Coding For Image Classification”. In: *IEEE International Workshop on Content-Based Multimedia Indexing*. 2014.
- [BRD13b] M. Babae, G. Rigoll, and M. Datcu. “Immersive Visualization Of The Quality Of Dimensionality Reduction”. In: *International Conference on Sensors & Models in Photogrammetry & Remote Sensing*. 2013.
- [Bab+14h] M. Babae et al. “Discriminative Feature Learning From SAR Images”. In: *Big Data from Space, Intenational Conference on*. 2014.

---

## Supervised Students' Theses

- [Kus15] R. Kusterer. "Virtuelle Erde: Eine immersive Visualisierung von Radarbildern". Bachelor Thesis. Technische Universität München, 2015.
- [Tso14] S. Tsoukalas. "Development And Evaluation Of Active Learning Algorithms For The SAR Dataset". Master's Thesis. Technische Universität München, 2014.
- [Yu15] X. Yu. "Immersive Interactive Dimensionality Reduction for Radar Images". Master's Thesis. Technische Universität München, 2015.