Received: 15 December 2021 | Revised: 10 October 2022 | Accepted: 14 December 2022

DOI: 10.1002/sim.9637

RESEARCH ARTICLE

Statistics in Medicine WILEY

A tree-based modeling approach for matched case-control studies

Gunther Schauberger[1] | **Luana Fiengo Tanaka**[1] | **Moritz Berger**[2]

[1]Chair of Epidemiology, Department of Sport and Health Sciences, Technical University of Munich, Munich, Germany

[2]Institute of Biomedical Statistics, Computer Science and Epidemiology, University of Bonn, Bonn, Germany

Correspondence
Gunther Schauberger, Chair of Epidemiology, Department of Sport and Health Sciences, Technical University of Munich, Munich, Germany.
Email: gunther.schauberger@tum.de

Funding information
German Research Foundation (DFG), Grant/Award Number: BE7543/1-1

Conditional logistic regression (CLR) is the indisputable standard method for the analysis of matched case-control studies. However, CLR is strongly restricted with respect to the inclusion of non-linear effects and interactions of confounding variables. A novel tree-based modeling method is proposed which accounts for this issue and provides a flexible framework allowing for a more complex confounding structure. The proposed machine learning model is fitted within the framework of CLR and, therefore, allows to account for the matched strata in the data. A simulation study demonstrates the efficacy of the method. Furthermore, for illustration the method is applied to a matched case-control study on cervical cancer.

KEYWORDS

CART, conditional inference trees, conditional logistic regression, matched case-control studies, matched pairs

1 | INTRODUCTION

Case-control studies are a popular tool to determine risk factors for specific diseases, especially if the disease is rare in the population of interest because it guarantees for a sufficient number of cases in the study. Another popular field of application is the case of disease outbreak investigation. In case-control studies, researchers select a group of cases and a group of controls, which are known to be free of the respective disease. Cases and controls are compared with respect to an exposure, which is usually collected retrospectively. Often, the analysis of a case-control study requires adjustment for potential confounders because the respective exposure was not assigned randomly. A well-known problem is that confounders may be very unbalanced between cases and controls if no restrictions were used for the selection of the controls. For example, for many types of cancer the average age of cases will be higher than in a randomly chosen control group. Also, younger people seem to be more willing to participate in biomedical research.[1]

Therefore, matched case-control studies are frequently used to guarantee balance with respect to important confounders.[2] In individual matching, each case is matched directly to a certain amount of controls specifically selected for this case. Among others, age and sex are popular matching factors. Controls can be recruited from different sources, including neighborhood/community, family, hospital/clinic.[3] These controls might share environmental and societal exposures with cases, thus one automatically accounts for potential confounders simultaneously that are difficult to measure on an individual level. Examples are regional or geographic differences, social status or environmental factors.[4]

There is an ongoing debate about common misconceptions regarding the benefit of matching itself and whether matched data always require a data analysis that accounts for the matching structure.[2,5] The aforementioned publications agree that matching can increase the efficiency by increasing the precision of the estimation of the exposure effect because cases and controls are well balanced with respect to potential confounders. However, matching can also have an adverse effect, namely to create selection bias. Selection bias, which can be removed by adjusting for the matching variables, is caused by the fact that matching makes cases and controls more similar, not only with respect to potential confounders but also with respect to the exposure. In an extreme scenario, all cases match their controls also with respect to the exposure of interest purporting the absence of an effect of the exposure. This is in contrast to the common misconception that matching on certain factors allows to ignore these factors in data analysis.[5]

Another strongly related and important question is whether the matching structure has to be considered in the statistical analysis. Commonly, this leads to the question whether conditional logistic regression (CLR), originally proposed by Breslow et al,[6] or unconditional logistic regression (ULR) is more appropriate.[7,8] Following the argumentation of Pearce,[5] matched case-control data do not necessarily have to be analyzed in a "matched analysis." Especially in cases of so-called "loose-matching," where matching is performed on only few rather unspecific factors like gender and age, it can be sufficient to account for these confounders as covariates while using ULR. However, if matching is performed on many factors or factors with a large number of possible values (eg, neighborhood), the matching should be accounted for in the statistical analysis. In these circumstances, CLR is the preferable choice because compared to ULR it helps to avoid the so-called sparse data problem.[9,10] In general, the sparse data problem occurs if for any statistical model an increasing number of observations is accompanied by an increase in the number of parameters, leading to inconsistent estimators. In matched case-control studies, it particularly occurs if matching is performed on many different factors or on factors with many possible values (eg, neighborhood), which would require a large number of adjustment parameters to be estimated in ULR. CLR can avoid this problem by conditioning on the respective strata and, thereby, eliminating the large number of parameters from the likelihood. CLR makes it unnecessary to account for exactly matched factors, while interval-matched factors still should be accounted for (see also Section 3.3). Furthermore, Wan et al[8] demonstrated that ULR is very sensitive to misspecifications of the functional form of the matched factors while CLR is much more robust. Therefore, even if a "matched" statistical analysis may not be required for each matched case-control study, it is a robust choice in most cases.

In recent years, some other alternatives to the classical CLR have been proposed. For example, Avalos et al[11] and Reid and Tibshirani[12] proposed regularization approaches using an L_1 penalty term, which can better deal with high-dimensional data and sparse exposure or confounding variables than classical CLR. Stanfill et al[13] proposed a pre-processing step for paired matched case-control data which allows to use a wide variety of machine learning approaches. However, this procedure is restricted to the special case of 1:1 matching. Zetterqvist et al[14] proposed the concept of doubly-robust CLR.

In this work, we propose an alternative method to the classical CLR when dealing with matched case-control-studies. The main idea is to combine CLR with the concept of tree-based classifiers. Instead of using a linear combination of the variables, partitions in the covariate space are determined in a data-driven manner and embedded into the CLR framework (also accounting for the matching strata) using indicator variables. The approach proposed in this manuscript differs from the previously mentioned approaches and from classical CLR as it does not require linearity for the associations between covariates and the binary outcome and automatically accounts for interactions between covariates.

Up to now, no statistical learning or machine learning approach has been proposed that goes beyond linear modeling of the covariate effects. In principle, there exists a variety of machine learning methods as possible candidates, such as nearest neighbors, Naive Bayes, neural networks, or bagging, only to name a few. Here, we make a first start using trees. The main challenge is the need to incorporate the specific data structure (the matched strata) into the learning method adequately. Second, as typically machine learning models are black boxes with a limited ability to interpret the effects of individual variables, it is reasonable to allow for a separate modeling of the exposure effect. Both challenges are solved by the approach proposed in this manuscript.

This manuscript is structured as follows: Section 2 describes the basic concept of CLR while Section 3 introduces the the newly proposed tree-based method for the analysis of matched case-control data. Section 4 extends the proposed tree framework by separating the exposure effect from the other covariates. The efficacy of the proposed method is illustrated by a simulation study in Section 5 and by an application to real matched case-control data in Section 6. Section 7 closes the article with a discussion and some concluding remarks.

2 | CONDITIONAL LOGISTIC REGRESSION

The basic CLR model has the form

$$\log\left(\frac{\mathrm{P}(y_{ij}=1)}{\mathrm{P}(y_{ij}=0)}\right)=\alpha_i+\mathbf{z}_{ij}^T\boldsymbol{\gamma}\quad i=1,\ \dots\ ,n,\ j=1,\ \dots\ ,m_i, \tag{1}$$

where $y_{ij}\in\{0,1\}$ denotes a binary outcome. The observations come in n clusters $s_i,\ i=1,\ \dots\ ,n$, of size m_i. The cluster structure is accounted for in the model via cluster-specific intercepts α_i. The linear predictor of the model further contains the linear term $\mathbf{z}_{ij}^T\boldsymbol{\gamma}$ with coefficient vector $\boldsymbol{\gamma}$ and covariate vector $\mathbf{z}_{ij}$.

In matched case-control studies, it holds that $y_{ij}=1$ for cases and $y_{ij}=0$ for controls and that the clusters are defined by the matching strata. The vector $\mathbf{z}_{ij}$ collects the exposure variable (in the following denoted by x) and all further covariates of observation j in cluster i. Due to the fact that the stratum-specific intercepts automatically inherit all effects of exactly matched variables, these are explicitly not included in $\mathbf{z}$. However, $\mathbf{z}$ may also contain interval matched variables, that is, where matching was not exact (frequently used for age-matching).

We assume each of the n strata to contain only one case, that is, $\sum_{j=1}^{m_i}y_{ij}=1$. For simplicity, in each stratum we assume the first observation to be the case, that is, $y_{i1}=1$ for $i=1,\ \dots\ ,n$. Commonly, parameter estimation in logistic models is done using the maximum likelihood approach. In CLR, the stratum-specific intercepts α_i are eliminated from the likelihood by conditioning on the number of cases per stratum. This leads to the conditional likelihood

$$L_c(\boldsymbol{\gamma})=\prod_{i=1}^{n}\frac{\exp(\mathbf{z}_{i1}^T\boldsymbol{\gamma})}{\sum_{j=1}^{m_i}\exp(\mathbf{z}_{ij}^T\boldsymbol{\gamma})},$$

and the corresponding conditional log-likelihood

$$l_c(\boldsymbol{\gamma})=\sum_{i=1}^{n}\left(\mathbf{z}_{i1}^T\boldsymbol{\gamma}-\log\left(\sum_{j=1}^{m_i}\exp(\mathbf{z}_{ij}^T\boldsymbol{\gamma})\right)\right).$$

See Breslow and Day[9] for further details. The maximum likelihood estimates are derived by maximizing the conditional log-likelihood. The conditional log-likelihood from CLR models equals the partial log-likelihood of stratified Cox models, where all event times are assumed to be equal to 1, cases are treated as observations experiencing their event and controls are treated as right-censored observations. Therefore, many software implementations use routines for fitting stratified Cox models to estimate CLR models.

3 | CONDITIONAL LOGISTIC REGRESSION TREES

The method proposed in this manuscript is based on the concept of classification trees, which is an established classification method in statistical or machine learning. They originate from the framework of classification and regression trees (CARTs) proposed by Breiman et al.[15] In CARTs, the covariate space is partitioned using recursive binary splits. In each partition, a constant is fitted (to obtain predictions or predicted probabilities). For classification tasks, the predicted probabilities can simply be derived from the relative frequencies of the single outcome categories among the observations which fall into the respective partition. CARTs are commonly visualized via hierarchical decision trees (see Figure 1 for an illustrative example).

3.1 | Algorithm

Ordinary classification trees cannot be applied to data from matched case-control data as they would not account for the matching structure. Therefore, we propose to implement the concept of recursive partitioning in the covariate space within the framework of CLR. Thus, we take advantage of the fact that CLR accounts for matching by conditioning on the number of cases per stratum. The main idea is to replace the linear predictor of model (1) by a tree structure. This tree structure is built up by sequential splits in the covariate space.

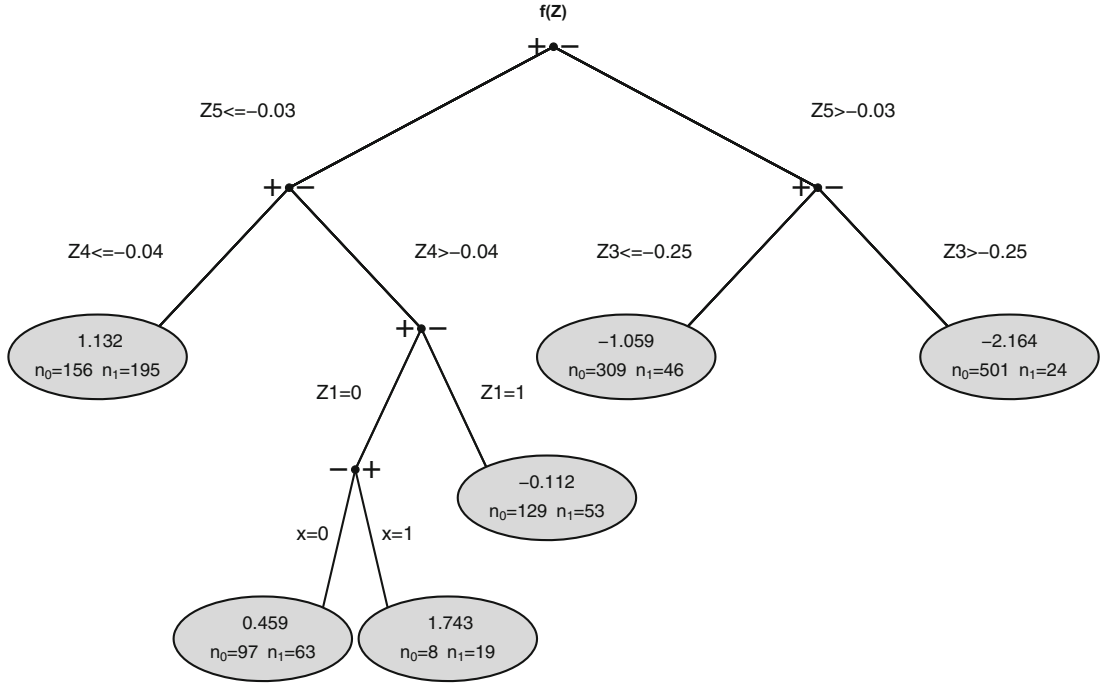


FIGURE 1 Estimated tree structure for illustrative data with confounding variables $Z_1, \ldots, Z_5$ and exposure variable X. Terminal nodes display the corresponding estimated parameters $\hat{\delta}_1, \ldots, \hat{\delta}_6$ and the numbers of cases n_1 and controls n_0

The initial model M_0 is the model only containing the strata-specific intercepts

$$M_0 : \quad \log\left(\frac{\mathrm{P}(y_{ij}=1)}{\mathrm{P}(y_{ij}=0)}\right) = \alpha_i.$$

Starting from the root node (containing all observations), we gradually search for binary splits in the covariates collected in $\mathbf{z}$ that further improve the fit of this model. A split divides the respective nodes into two sub-nodes, and is incorporated into the model using a corresponding indicator variable. For a metric (both continuous and discrete) or ordinal variable, the model after the first split at threshold c in variable z_k has the form

$$\log\left(\frac{\mathrm{P}(y_{ij}=1)}{\mathrm{P}(y_{ij}=0)}\right) = \alpha_i + \delta I(z_{ijk} \leq c),$$

where $I(\cdot)$ is the indicator function. Step by step, the variable and the split point are chosen that lead to the highest improvement of the conditional log-likelihood of the model. After termination of the algorithm according to an appropriate stopping criterion (see below), the final model can be denoted as

$$\log\left(\frac{\mathrm{P}(y_{ij}=1)}{\mathrm{P}(y_{ij}=0)}\right) = \alpha_i + f(\mathbf{z}_{ij}),$$

where $f(\mathbf{z}_{ij})$ describes the effect of the variables collected in $\mathbf{z}$ represented by a tree. With $S_1, \ldots, S_t$ representing the terminal nodes of the tree, $f(\mathbf{z}_{ij})$ can be denoted as

$$f(\mathbf{z}_{ij}) = \delta_1 I(\mathbf{z}_{ij} \in S_1) + \cdots + \delta_t I(\mathbf{z}_{ij} \in S_t). \tag{2}$$

All nodes are determined by a product of indicator functions. For example, if the splits were in the metric variables z_1 and z_4 a node may be determined by $I(\mathbf{z}_{ij} \in S) = I(z_{ij1} > 20)\, I(z_{ij4} \leq 10)$.

In all tree-based methods, one has to decide in particular how to split and how to determine the size of the trees. In traditional approaches, one typically grows large trees and prunes them to an adequate size afterwards, see Breiman

et al[15] and Ripley.[16] An alternative strategy, which was propagated within the conditional unbiased recursive partitioning framework,[17] is to directly control the size of the trees by early stopping. We also use this approach and control the necessity of splits by using tests based on the conditional log-likelihood of the models.

To search for the best split among all the variables and the set of possible split points one examines all the null hypotheses $H_0 : \delta = 0$ and $H_1 : \delta \neq 0$ (of the newly added parameter) and selects the split with the maximal LR test statistic (minus two times the difference in the conditional log-likelihood values). This corresponds to selecting the model with minimal deviance, which is also equivalent to minimizing the entropy.

To decide whether the selected split should be performed, we apply a concept based on maximally selected statistics. The basic idea is to investigate the dependence of the binary outcome and the selected variable at a global level that takes the number of splits into account. For one fixed variable z_k, one simultaneously considers all LR test statistics T_{kc_k}, where c_k are from the set of possible split points, and computes the maximal value statistic $T_k = \max_{c_k} T_{kc_k}$. The P-value that can be obtained from the distribution of T_k provides a measure for the relevance of variable z_k. Importantly, the result is not influenced by the number of split points, therefore the method explicitly accounts for the involved multiple testing problem; for similar approaches, which inspired the proposed method, see Hothorn and Lausen,[18] Shih,[19] Shih and Tsai,[20] Strobl et al.[21] As the distribution of T_k is unknown we use a permutation test to obtain a decision on the null hypothesis. The distribution of T_k is determined by computing the maximal value statistics based on random permutations of variable z_k. A random permutation of variable z_k breaks the relation of the variable and the outcome in the original data. By computing the maximal value statistics for a large number of permutations one obtains an approximation of the distribution under the null hypothesis and the corresponding P-value. Given an overall significance level α the significance level for the permutation test that tests splits in one variable is chosen by α/p, where p denotes the number of covariates. To determine the P-value with sufficient accuracy, the number of permutations should increase with the number of variables.

Altogether, the following steps are carried out during the fitting procedure:

1. *(Initial model)*. Fit model M_0.
2. *(Tree building)*.
   (a) For all variables z_k, $k = 1, \ldots, p$, fit all the candidate models with one additional split in one of the already built nodes.
   (b) Select the best model with the maximal LR test statistic.
   (c) Carry out the permutation test for the selected node (defined by a combination of variable and split point) using the maximal value statistic with significance level α/p. If significant, fit the selected model and continue with Step 2(a), else continue with Step 3.
3. *(Selected model)*. Fit the final model with components $\alpha_i, \delta_1, \ldots, \delta_t$.

It should be noted that an additional split also has an effect on all remaining terminal nodes because all parameter estimates change if an additional split is performed. This is in contrast to the way trees are grown in traditional recursive partitioning where the remaining terminal nodes are not affected by a new split.

The threshold α for the permutation tests is the main tuning parameter for the proposed method. However, we found the method to be rather robust for different choices of this parameter (see also Reference 22) and if not stated otherwise we use a default value of $\alpha = 0.05$. To avoid the problem that $2^{K-1} - 1$ possible splits need to be considered when splitting in a nominal variable with K categories, we apply the sorting algorithm proposed by Wright and König.[23] This means that we order the categories once prior to tree building and subsequently treat the variable as ordinal.

An alternative to the use of permutation tests is to consider an information criterion, like the BIC, for early stopping. In this case a large tree is grown first by repeating Steps 2(a) and (b) of the fitting procedure. Second the optimal sub-tree is selected according to the minimal BIC. A comparison of the two pruning procedures is shown in the simulation section.

As shown in Equation (2), the tree structure is incorporated into the formula of CLR via indicator variables. The parameters $\delta_1, \ldots, \delta_t$ accompanying these indicator variables are identifiable only up to a constant location shift. Therefore, in estimation we have to apply a side constraint. We can either choose a reference node and restrict the corresponding parameter to zero (eg, $\delta_t = 0$) or we can impose the symmetric side constraint $\sum_{o=1}^{t} \delta_o = 0$. In either case, the interpretation of the node parameters $\delta_1, \ldots, \delta_t$ in absolute size is not meaningful. Interpretation should always solely be based on pairwise differences between node parameters. For example, $\exp(\delta_1 - \delta_2)$ represents the odds ratio when comparing observations from node 1 compared to observations from node 2.

The tree algorithm may lead to partitions in the covariate space where perfect discrimination between cases and controls is reached. This is desirable in general as it is the explicit goal of tree methods to separate the different classes.

However, perfect separability is a problem for parameter stability within the estimation framework of CLR. Therefore, we allow for an additional refitting step after the final tree has been determined. This refitting step employs regularization using a small L_2 penalty to stabilize the tree parameter estimates $\delta_1, \ldots, \delta_t$. The penalty term contains the corresponding squared parameter values. It is added to the regular conditional likelihood $l_c(\cdot)$ and is controlled by the tuning parameter λ. Optimization is then applied to the penalized likelihood

$$l_p(\cdot) = l_c(\cdot) + \lambda \sum_{o=1}^{t} \delta_o^2, \tag{3}$$

instead of $l_c(\cdot)$. As the penalty is only intended to stabilize estimation of the tree parameters, the tuning parameter λ can simply be set to a small value and does not require a systematic tuning procedure. A standard value we used was $\lambda = 10^{-20}$.

3.2 | Illustrative example

For illustration, the method is applied to a simulated data set. It is taken from setting B of the simulation study presented in Section 5. The data set contains 400 strata with 1 case and 3 controls each. It contains five confounding variables $Z_1, \ldots, Z_5$ (where Z_1 and Z_2 are binary and $Z_3, \ldots, Z_5$ are continuous) and one binary exposure variable X. Figure 1 depicts the estimated tree structure for the illustrative data.

We use the symmetric side constraint where $\sum_{o=1}^{5} \hat{\delta}_o = 0$. We can see that the first split in the covariate space is in Z_5 with split point -0.03. The resulting node $Z_5 > -0.03$ is split further with respect to the exposure variable Z_3. The node $Z_5 \leq -0.03$ is split further with respect to variable Z_4, for $Z_4 > -0.04$ with respect to variable Z_1, and for $Z_1 = 0$ in the exposure variable X. Finally, the tree ends up in six terminal nodes, leading to six parameter estimates $\hat{\delta}_1, \ldots, \hat{\delta}_6$. Beside the estimated parameters $\hat{\delta}_i$, the various terminal nodes also contain information about how many cases (n_1) and how many controls (n_0) fall into the respective node. As further additional information, + or − signs indicate for each split point, which sub node leads to increased (+) or decreased (−) odds. The estimated parameters $\hat{\delta}_i$ and the +/− signs associated with the first split indicate, that $Z_5 > 0.05$ decreases the odds as the corresponding estimated parameters are negative and smaller than the set of estimated parameters corresponding to $Z_5 \leq -0.03$. The highest odds appear among observations where $Z_5 \leq -0.03$, $Z_4 > -0.04$, $Z_1 = 0$ and $X = 1$.

In this example, there is no special treatment (see Section 4) of the exposure effect. We can see, that in the estimated tree the exposure status only makes a difference for a small subgroup. In this subgroup, being exposed clearly increases the odds of having the disease.

3.3 | Interval matching

The case of interval-matched variables requires special attention. A typical example of a matching variable, for which interval matching is used, is age. Typically, the age of cases and their matched controls has to be from the same age category (eg, 50-55 years). As mentioned before, in contrast to exactly matched variables interval-matched variables must be adjusted for in the analysis. We suggest to use the so-called residual age[2] as a covariate. Residual age can be calculated in different ways, for example as the individual difference between a persons age and the mean of the corresponding age category or the mean of the controls within each stratum. In each case, the same value is subtracted from all age values within one stratum.

In CLR, using residual age or the original age variable gives equivalent results because the strata-specific intercepts α_i anyway capture the average age effect in a stratum. However, for our tree-based approach the results differ between both approaches. For real age, the splits will assign all subjects to the same partition for almost all strata while for residual age the splits will split within the strata in almost all cases. As an example, let us assume three strata I, II, and III with arbitrary values of age and accompanying values of residual age as displayed in Table 1.

We can see, that splitting age at some specific value (eg, at 64 years) will automatically put all observations from one stratum into the same partition for at least two of the three strata. Splitting in residual age (eg, at 0), will create a split within the stratum for all three strata. Therefore, these two methods are clearly not equivalent here and it is essential to transform interval-matched variables into residual differences when they are supposed to be accounted for in the model.

TABLE 1 Illustrative example for comparing the different effects of including age or residual age into the tree-based approach for further age adjustment additional to age matching

	Stratum I		Stratum II		Stratum III	
	Age	**Residual age**	**Age**	**Residual age**	**Age**	**Residual age**
Case	24	1	45	2	63	0
Control 1	21	−2	44	1	62	−1
Control 2	25	2	42	−1	61	−2
Control 3	22	−1	43	0	65	2

4 | SEPARATION OF THE EXPOSURE EFFECT

Most matched case-control studies aim at quantifying the effect of one particular of the covariates, the exposure variable x. In this case, we are specifically interested in the association (or rather the causal effect) of an exposure x on the outcome. The exposure variable is binary with $x \in \{0, 1\}$ where $x = 1$ for exposed and $x = 0$ for non-exposed persons. All the other variables that are supposed to be included in the analysis are collected in the vector $\mathbf{z}$ (which no longer contains x). In the following, we will elaborate on how to separate the exposure effect from the remaining covariates in order to have a separate global estimate of the exposure.

4.1 | Algorithm

To clarify the special role of the exposure x_{ij}, we now denote model (1) as

$$\log\left(\frac{\mathrm{P}(y_{ij}=1)}{\mathrm{P}(y_{ij}=0)}\right) = \alpha_i + x_{ij}\beta + \mathbf{z}_{ij}^T\boldsymbol{\gamma} \quad i = 1, \ldots, n,\ j = 1, \ldots, m_i, \tag{4}$$

where α_i represents a stratum-specific intercept, β is the regression coefficient (interpretable as the adjusted log odds-ratio) of the exposure and $\boldsymbol{\gamma}$ denotes all further covariate effects. The conditional log-likelihood for model (4) has the form

$$l_c(\beta, \boldsymbol{\gamma}) = \sum_{i=1}^{n}\left(x_{i1}\beta + \mathbf{z}_{i1}^T\boldsymbol{\gamma} - \log\left(\sum_{j=1}^{m_i}\exp(x_{ij}\beta + \mathbf{z}_{ij}^T\boldsymbol{\gamma})\right)\right).$$

In the proposed tree-based model the linear term $\mathbf{z}_{ij}^T\boldsymbol{\gamma}$ again is replaced by a tree structure while at the same time the parametric structure of the CLR model (4) is preserved, which keeps the exposure effect easily interpretable. For that purpose, the starting model M_0 is extended by a parametric exposure effect:

$$M_0: \quad \log\left(\frac{\mathrm{P}(y_{ij}=1)}{\mathrm{P}(y_{ij}=0)}\right) = \alpha_i + x_{ij}\beta.$$

Given this model (ie, given the initial parameter estimate $\hat{\beta}$), we search for binary splits in the variables contained in $\mathbf{z}$. Accordingly, the proposed model can be denoted as

$$\log\left(\frac{\mathrm{P}(y_{ij}=1)}{\mathrm{P}(y_{ij}=0)}\right) = \alpha_i + x_{ij}\beta + f(\mathbf{z}_{ij}),$$

where $f(z_{ij})$ describes the effect of the covariates represented by a tree. With $S_1, \ldots, S_t$ representing the terminal nodes of the tree, $f(\mathbf{z}_{ij})$ can again be denoted as

$$f(\mathbf{z}_{ij}) = \delta_1 I(\mathbf{z}_{ij} \in S_1) + \ldots + \delta_t I(\mathbf{z}_{ij} \in S_t).$$

The main advantage of this model (from now on referred to as *CLogitTree*) over ordinary CLR (see Equation (4)) is its flexibility. In particular, no linearity of effects is assumed and complex interaction effects between covariates are found completely data-driven. Also in the model with a separate exposure effect we allow to use an L_2 penalty term as described in Equation (3). The exposure parameter β is not included into the penalty term, but nevertheless may be affected indirectly by the penalty.

The model described above can be seen as a hybrid between a classical regression model and a tree, where the linear predictor of a regression model consists both of classical linear terms and a tree component. Similar attempts have been made before in different settings. Among others, Su et al[24] use hybrid models to check the assumptions in classical linear regression while Su and Tsai[25] proposed tree-augmented Cox regression. A somewhat related concept was proposed by Zeileis et al[26] where recursive partitioning is used to find subspaces in the covariate space, in which separate models are fitted.

4.2 | Confidence intervals

For the separated exposure effect it is desirable to have confidence intervals beside the point estimate. However, we cannot directly use the confidence intervals provided by the underlying conditional logistic model as it will not take the selection process into account when building the tree. Accordingly, these confidence intervals would underestimate the variability of the point estimate. However, confidence intervals can be derived via a nonparametric bootstrap approach. For a total of B bootstrap samples, the strata are treated as the observation units as we do not want to break up matched strata. Therefore, a single bootstrap data set is constructed by sampling n strata with replacement from the original data, where n is the total number of strata in the original data. For each bootstrap sample, we compute a *CLogitTree* model and save the corresponding estimate for the exposure effect. This procedure provides an estimate of the empirical bootstrap distribution of the exposure effect. From the B estimates we calculate the corresponding empirical quantiles as the boundaries of the confidence interval. The bootstrap confidence intervals for the exposure effect are computationally demanding and, therefore, rather time-consuming.

4.3 | Illustrative example

We adapt the illustrative example from Section 3.2 to an analysis where the exposure variable x is treated separately. Now, the estimated model consists both of an estimated log exposure effect and an estimated tree (see Figure 2). The respective point estimate and 95% confidence interval of the exposure effect is $\exp(0.619) = 1.858$ (95% CI 1.320 – 3.045) for *CLogitTree* and $\exp(0.4833) = 1.621$ (95% CI 1.155 – 2.275) for CLR. In the data generating process (DGP), the true log exposure effect was $\log(2) = 0.693$. Accordingly, we can see that the estimate from *CLogitTree* $\exp(\hat{\beta}) = 1.858$ is slightly closer to the true value of 2 than the CLR estimate $\exp(\hat{\beta}) = 1.621$. Indeed, in the corresponding simulation scenario (ie, scenario B in Section 5), we will see that, in contrast to the CLR estimates, the exposure effect estimates from *CLogitTree* are unbiased. The confidence interval from *CLogitTree* is wider than the confidence interval from CLR, which may simply be caused by the fact that for CLR the confidence interval is estimated via a closed formula while it is estimated via bootstrap for *CLogitTree*.

The tree depicted in Figure 2 shows that (compared to the true tree depicted in Figure 3B) the first split is chosen almost perfectly correct ($Z_5 > -0.03$ instead of $Z_5 > 0$).

The first three splits exactly equal the first splits in the model without a separate exposure effect (compare Figure 1). The last split (for $Z_5 \leq -0.03$, $Z_4 > -0.04$, $Z_1 = 0$) is missing compared to Figure 1, which is not surprising as now the exposure status x is not included in the algorithm to find split points but is modeled separately. Overall, almost all splits are found correctly, but the last two splits in Z_2 are missing compared to the DGP (compare Figure 3B).

5 | SIMULATION STUDY

In a simulation study, we want to explore how the method works and how well it performs compared to CLR. We explore five different settings for the DGP, which will be named setting A, setting B, setting C, setting D, and setting E. We first

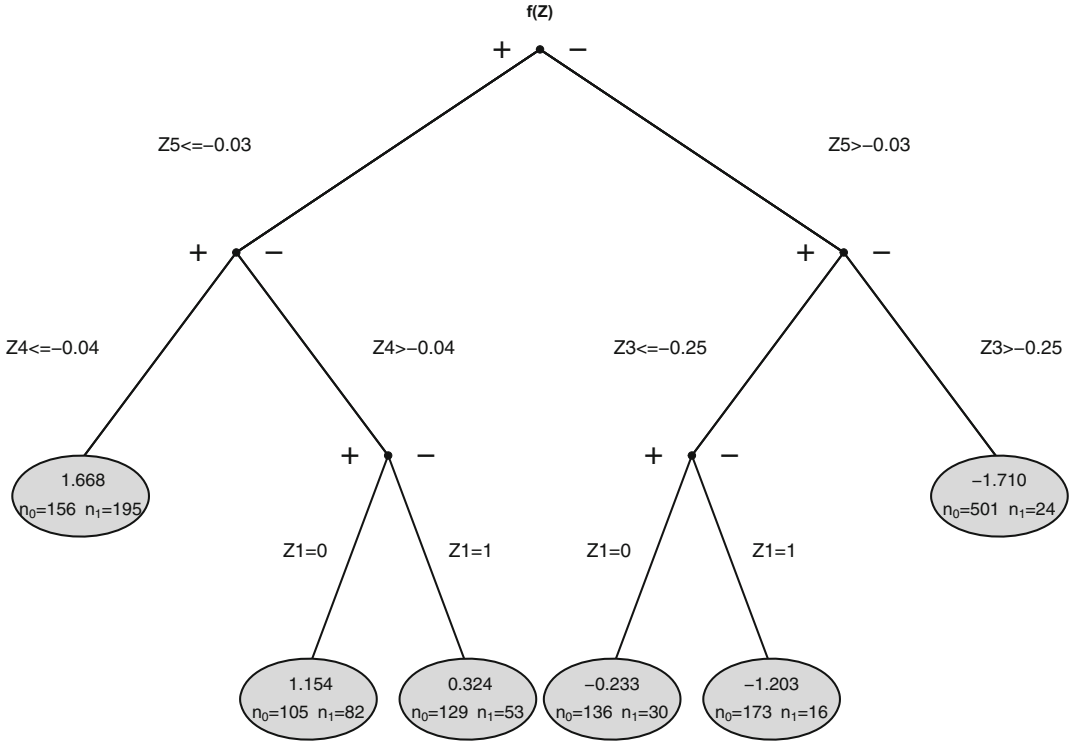


FIGURE 2 Estimated *CLogitTree* for illustrative data with a separated exposure effect estimate. For all terminal nodes, the respective parameter estimates and the numbers of cases n_1 and controls n_0 are displayed

present the design of the simulation study describing the DGP in the different settings before we present the respective results.

5.1 | Simulation settings

For all settings, we first created a hypothetical population consisting of 500 000 persons, which were randomly distributed to 1000 districts. Also, five person-specific characteristics (Z_1 to Z_5) were generated as covariates. We use a five-dimensional standard normal distribution and dichotomize variables Z_1 and Z_2 with a threshold of 0. For each person we generated a personalized probability to be exposed as well as an individual probability to suffer from the respective disease.

First, a binary exposure variable X was sampled. The probability of being exposed $P(X = 1)$ was generated with the same DGP for all five settings. For person i in district j we define the linear predictor

$$\eta_{ij}^X = -2 + \tau_j^X + 0.3 \cdot Z_{1i} - 0.3 \cdot Z_{2i} - 0.4 \cdot Z_{3i} + 0.3 \cdot Z_{4i} + 0.2 \cdot Z_{5i},$$

where τ_j^X is a specific district effect (randomly drawn from a uniform distribution between −2 and 2). We use η_{ij}^X to create a person-specific probability of being exposed as

$$P(X_{ij} = 1) = \frac{\exp(\eta_{ij}^X)}{1 + \exp(\eta_{ij}^X)}.$$

In the next step, for each person a probability of suffering from the disease was determined. This probability was defined as

$$P(Y_{ij} = 1) = \frac{\exp(\eta_{ij})}{1 + \exp(\eta_{ij})},$$

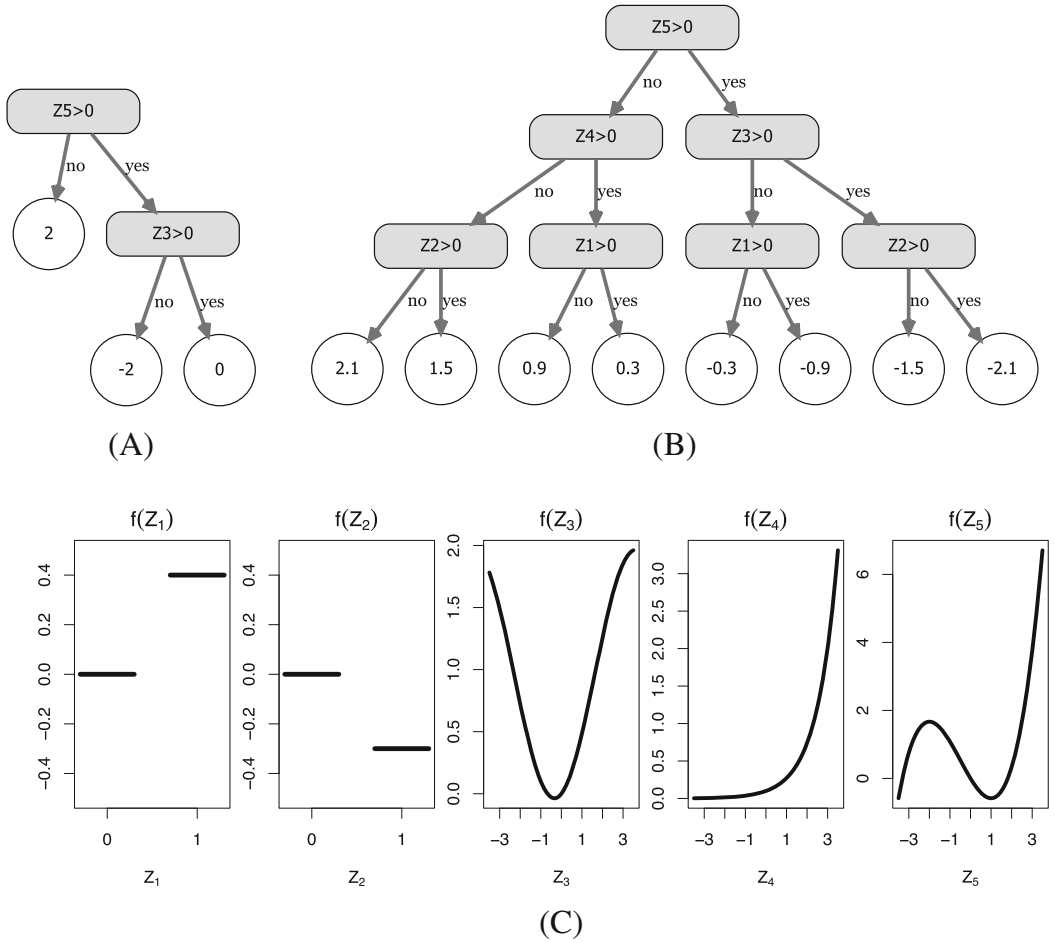


FIGURE 3 Functional terms $f(\mathbf{Z}_i)$ used in data generating process of simulation settings A, B, and D. (A) $f(\mathbf{Z}_i)$ in Setting A; (B) $f(\mathbf{Z}_i)$ in Setting B; (C) $f(\mathbf{Z}_i)$ in Setting D

with the linear predictor

$$\eta_{ij} = -2 + \tau_j + \beta \cdot X_i + \phi f(\mathbf{Z}_i).$$

Here, τ_j is again a district effect (randomly drawn from a uniform distribution between -2 and 2), β is the exposure effect and $f(\mathbf{Z}_i)$ represents a functional term based on the vector $\mathbf{Z}_i = (Z_{1i}, Z_{2i}, Z_{3i}, Z_{4i}, Z_{5i})$. This functional term $f(\mathbf{Z}_i)$ differed between the simulation settings. The exposure effect was set to $\beta = \log(2)$ for all settings, which corresponds to a true (adjusted) odds ratio of 2 as the exposure effect. The parameter ϕ is used to control the signal-to-noise ratio in the respective simulation runs. For each setting, it will be set to three different values $\phi = 0.5$ (low), $\phi = 1$ (medium), and $\phi = 2$ (high).

In settings A and B, the functional term $f(\mathbf{Z}_i)$ represents a decision tree. Setting C is the so-called "null"-setting, where variables Z_1 to Z_5 have no effect on the disease status. In setting D, the functional term consists of adding up linear terms for Z_1 and Z_2 and smooth terms for variables Z_3 to Z_5. The functional terms used in settings A, B, and D are displayed in Figure 3. In setting E, the functional term is defined via additive linear terms

$$f^E(\mathbf{Z}_i) = -0.4 \cdot Z_{1i} + 0.2 \cdot Z_{2i} - 0.3 \cdot Z_{3i} - 0.2 \cdot Z_{4i} + 0.4 \cdot Z_{5i}.$$

To create the final matched case-control data, we randomly sampled $n = 400$ persons from the diseased persons as our cases. For each case and restricted to the non-diseased persons from the same district as the case, we randomly sampled three controls, leading to matched strata of size $m_i = 4$. Accordingly, our final data set consisted of 400 strata and a total of 1600 observations.

5.2 | Simulation results

In the simulation study, we apply the proposed method *CLogitTree* including a separate exposure effect. As the main results of the simulations we present how well the exposure effect has been estimated and compare the results to conventional CLR. Furthermore, the methods are compared via the predictive conditional likelihood. For that purpose, in each run of each setting an additional validation data set is generated with the same size and structure as the training data. For each method, the (predictive) conditional likelihood is computed per stratum and averaged across strata. *CLogitTree* is used with three different significance levels (0.01, 0.05, and 0.1) as well as using BIC as pruning criterion for comparison.

For all settings, Figure 4 displays the parameter estimates of the log exposure effect. The dashed red line represents the true log exposure effect. The corresponding squared errors can be found in Figure S1 (see supplementary materials).

Overall, we can see that *CLogitTree* is rather robust against the choice of the pruning strategy, the performance based on permutation tests is very similar for all three values of α and also comparable if the optimal tree is selected via BIC. For setting A and setting B, we can see that for CLR the bias of the estimate increases with an increasing signal-to-noise ratio while it is nearly unbiased for *CLogitTree*. Therefore, we see here that if the DGP follows a tree structure *CLogitTree* outperforms CLR with respect to the estimation of the exposure effect. For setting C, as expected both methods work equally well. As this is the null-setting, we cannot apply different signal-to-noise ratios. Also in setting D, where the DGP contains additive nonlinear effects for the continuous variables and additive effects for the binary variables, no major differences are observed. Only for a high signal-to-noise ratio CLR shows some bias. In setting E, where the assumptions of CLR are fulfilled,*CLogitTree* is slightly biased, in particular for high increasing signal-to-noise ratio, while CLR is unbiased in all cases.

Figure 5 displays boxplots of the predictive conditional likelihood averaged over all strata for all settings. For all methods, a higher signal-to-noise ratio leads to higher values of the likelihood. In settings A, B, and D, *CLogitTree* outperforms CLR in terms of the conditional likelihood, whereas it performs equally well in setting C and worse in setting E. While it could be expected that the methods which can exactly reproduce the GDP of the respective setting (see settings A, B, and E) perform better, it is interesting to see that in setting D *CLogitTree* outperforms CLR. Thais indicates that the non-linear GDP from setting D was recovered better by *CLogitTree* than by CLR. The different versions of *CLogitTree* perform comparably well, with a slight advantage for the version with the BIC.

In terms of computation time (compare Figure S2 in the supplementary materials) *CLogitTree* is clearly more demanding compared to regular CLR (which usually took less than 0.1 s). In the simulation study, the permutation tests were parallelized over 20 cores which made the version based on permutation tests almost as fast as the BIC based version of *CLogitTree* in some settings.

The objective of this simulation study was to analyze the precision of the estimation of the exposure effect, which is usually the most important value when matched case-control data are analyzed. The exposure effect is estimated in an unbiased manner only if the confounding effect of the remaining variables is modeled sufficiently accurate. The analysis of the predictive conditional likelihood revealed that *CLogitTree* performed better or equal to CLR unless the GDP was strictly linear and additive. Overall, this small simulation study showed, that for a truly tree-structured DGP *CLogitTree* can outperform the classical approach of CLR in terms of bias and mean squared error of the respective exposure effect. In the additive nonlinear setting, both methods performed similarly well, while CLR outperformed *CLogitTree* in the additive linear setting. The differences mostly become visible for high signal-to-noise ratios while all methods perform comparably well for low signal-to-noise ratios.

6 | APPLICATION TO TEQAZ STUDY DATA

For further illustration, in this section the proposed method is applied to a real data example. The data originate from the so-called TeQaZ study,[27] a case-control study on cervical cancer. The main focus of the study was to examine the effect of frequent participation in cervical cancer screening (CCS) on the risk of cervical cancer. The exposure variable *CCS* was defined as *frequent participation*, if women had attended CCS at least every 3 years within the past 10 years, including at least once in the 3 years preceding diagnosis. *Frequent participation* was coded as $CCS = 0$ (ie, used as reference category) while infrequent participation was coded as $CCS = 1$. Matching was done using age and residence area where controls where matched to cases only if they live in the same area and if their age was at maximum 2 years younger or older than the age of the corresponding case. To account for residual age differences we calculated the variable *Age.Diff* defined as the difference between each individuals age and the average age in the respective stratum.

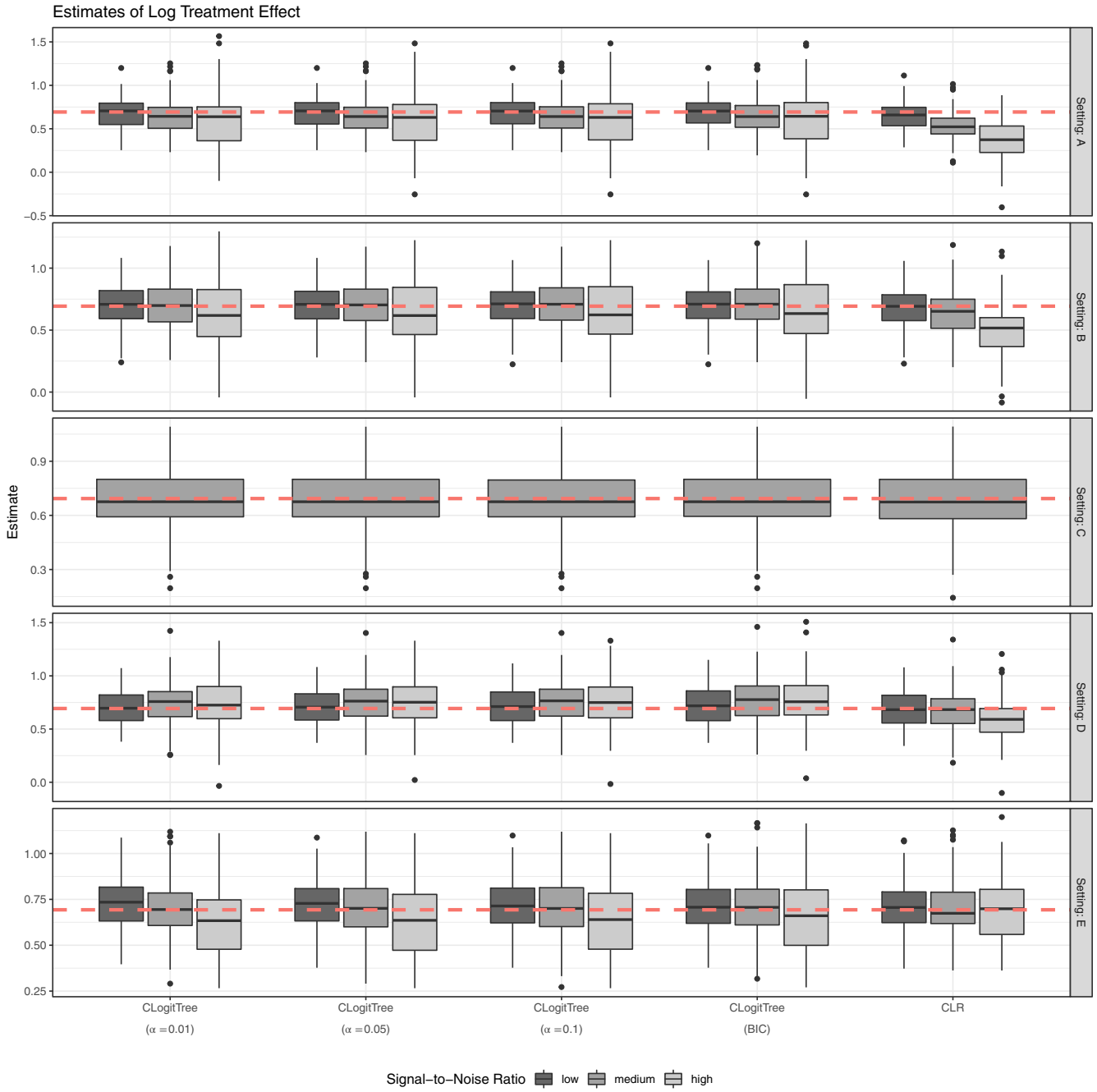


FIGURE 4 Parameter estimates for log exposure effect in different simulation settings, separately for CLR and different versions of *CLogitTree* and separately for different signal-to-noise ratios. Dashed red line represents the true log exposure effect

Table S1 (see supplementary materials) shows the list of variables contained in the corresponding data set. Tanaka et al[27] analyzed these data using CLR incorporated into a multiple imputation process. In contrast to this original analysis, we use only complete observations without any missings in the variables listed in Table S1. Therefore, while Tanaka et al[27] had used 217 cases and 649 controls our data set comprises 170 cases and 425 controls.

Tanaka et al[27] mostly used dichotomized versions of the original variables, which are also described in Table S1. In *CLogitTree*, dichotomization of variables in advance of the data analysis is not necessary as the variable levels are automatically grouped within the tree-building process in a data-driven manner. We will compare the results from CLR using the variables with coding scheme "Coding (CLR)" from Table S1 to the results from *CLogitTree* using the original coding of the variables. The results from ordinary CLR can be found in Table S2 (see supplementary materials).

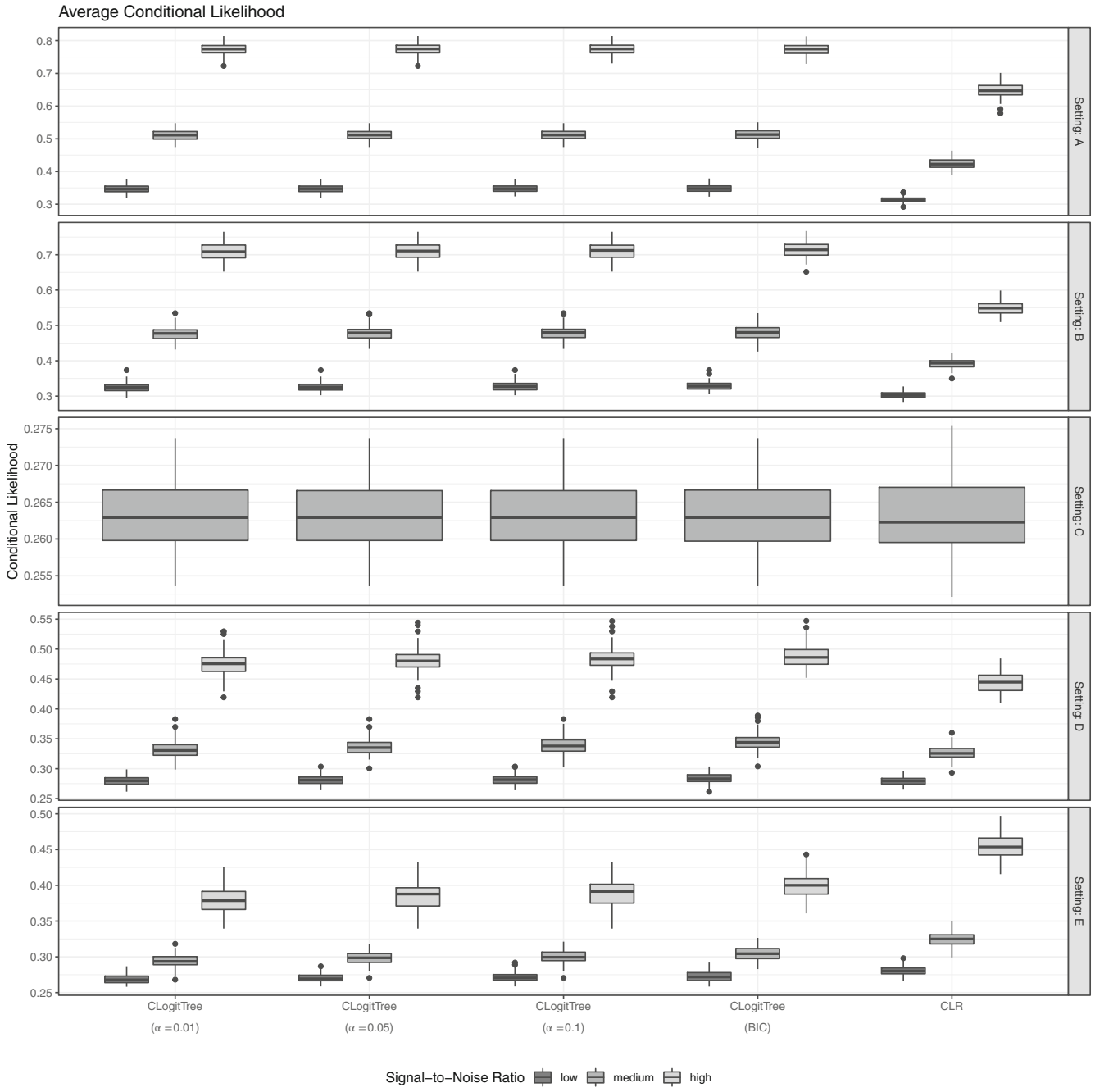
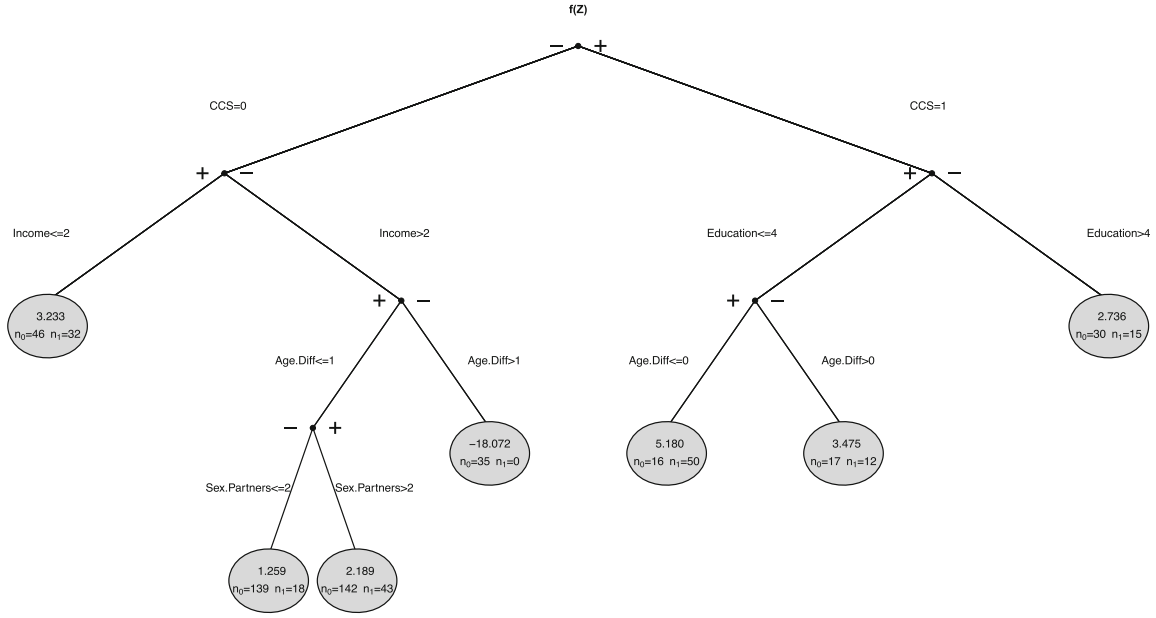


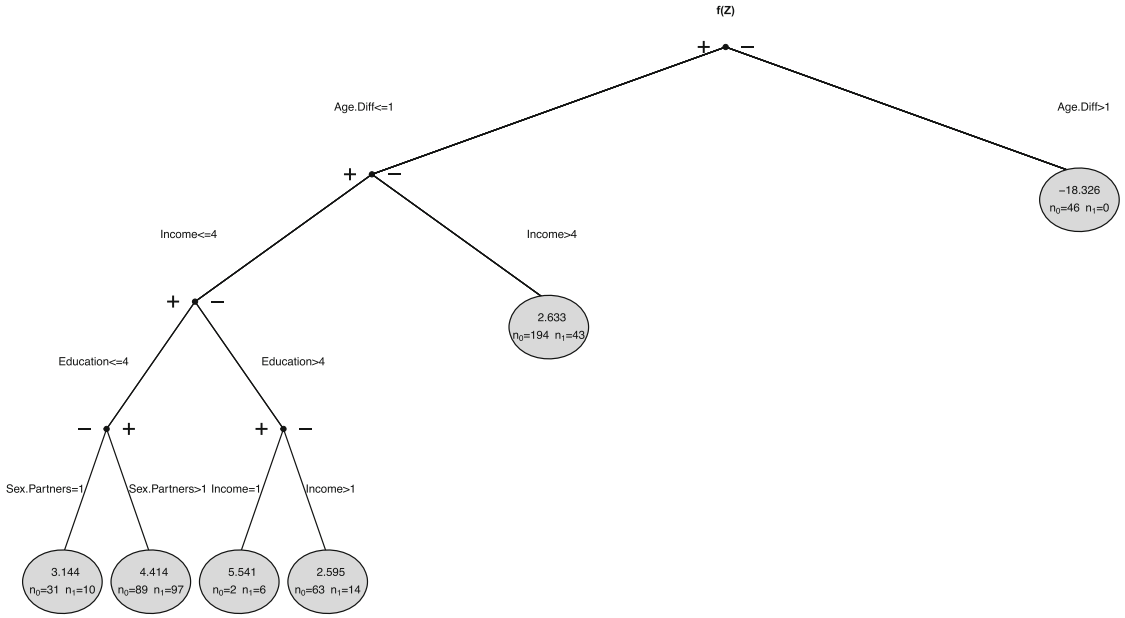
FIGURE 5 Average predictive conditional likelihood in different simulation settings, separately for CLR and different versions of *CLogitTree* and separately for different signal-to-noise ratios

Before applying the methods to the complete data set, a 10-fold cross-validation was performed on the data in order to compare the predictive performance of the models. Similar to the simulation study, we considered the predictive conditional likelihood for the comparison of the methods. The average predictive conditional likelihood was 0.50 for *CLogitTree* with $\alpha = 0.05$, 0.52 for *CLogitTree* using the BIC, 0.41 for a simple CLR using CCS as the only predictor, and 0.53 for ordinary CLR. We can see that CLR and *CLogitTree* using the BIC perform comparably good, and that the version using the BIC performs better than the version using permutation tests with threshold $\alpha = 0.05$. Therefore, in the following we will only present the results for the version with BIC.

CLogitTree will be applied to the data both in its simple version (without a separate exposure effect) and with a separate exposure effect for the exposure variable *CCS*. Both trees are displayed in Figure 6.



(A)



(B)

FIGURE 6 *CLogitTrees* for TeQaZ data for the model **without** (A) and **with** (B) a separate CCS exposure effect. For all terminal nodes, the respective parameter estimates and the numbers of cases n_1 and controls n_0 are displayed. (A) No separate CCS exposure effect; (B) separate CCS exposure effect

Interestingly, in Figure 6A the first chosen split is in the exposure variable *CCS*, where infrequent participation (ie, $CCS = 1$) is associated to higher odds for cervical cancer than frequent participation. Further splits are done in the variables *Education* (for $CCS = 1$) and *Income* (for $CCS = 0$). While the chosen split in *Education* is equal to the split used to dichotomize this variable in Tanaka et al,[27] this is not the case for *Income*. For $CCS = 0$ and $Income > 2$, we get another split in $Age.Diff > 1$. This split leads to a perfect separation of cases and controls in the corresponding terminal node with 35 controls and no cases. Therefore, the corresponding parameter estimate tends to $-\infty$ and, due to the symmetric side constraint, leads to rather big parameter estimates in the other terminal nodes. However, as described above we are mainly interested in the pairwise differences between the single estimates. These are very big for all differences

where the terminal node with perfect discrimination is involved, but in a reasonable magnitude for all other comparisons. Further splits are chosen in *Age.Diff* > 0 for *CCS* $= 1$ and *Education* > 4 and in *Sex.Partners* > 2 for *CCS* $= 0$, *Income* > 2 and *Age.Diff* ≤ 1.

Figure 6b shows the resulting tree of the model where CCS is incorporated in a separate exposure effect. Here, the first split is done for *Age.Diff* > 1, again leading to a terminal node with perfect separation and a very small parameter estimate. This split however is very hard to interpret as age was already a factor used for matching. Therefore, we simply see this split as a further age adjustment additional to the effect of age-matching accounting for potential residual age differences. For *Age.Diff* ≤ 1 (the node where most of the observations fall into), further splits are performed in *Income*, *Education* and *Sex.Partners*, leading to an interaction between these variables. Similar to the results from CLR ($\hat{\gamma}_{Income} = -0.856$ and $\hat{\gamma}_{Education} = -0.935$), we see in general that higher income and educational level seem to act as protective factors. However, in *CLogitTree* we see an additional effect of *Education* only for lower incomes. The majority of the variables included in CLR is not used for splitting in *CLogitTree*.

The parameter estimates presented in Figure 6B contain an extreme estimate due to the perfect separation in the respective terminal node. Perfect separation is desirable for our method from a tree perspective while it can be problematic in a regression setting where our tree-method is embedded. A simple solution to prevent parameter estimates tending to $\pm\infty$ is to apply L_2 penalization for the estimation of the tree parameters (as described in Section 3.1). Figure S3 (see supplementary materials) shows the corresponding result where a very small L_2 penalty (tuning parameter $\lambda = 10^{-20}$) has been applied in the estimation process.

In absolute size, the parameter estimate associated to the terminal node with perfect separation is much smaller now. Accordingly, also all pairwise differences with this parameter have decreased drastically. All other pairwise differences remain almost unchanged. For example, the difference in the estimates between *Sex.Partners* $= 1$ and *Sex.Partners* ≥ 1 in the subsample *Age.Diff* ≤ 1, *Income* ≤ 4 and *Education* ≤ 4 is $3.144 - 4.414 = -1.270$ in Figure 6B while it is $0.311 - 1.584 = -1.273$ in Figure S3.

Finally, Table S3 (see supplementary materials) contains the estimated exposure effects and the corresponding estimates for 95% confidence intervals for *CLogitTree*, *CLogitTree* with L_2 penalty, CLR and CLR^0. CLR^0 represents the CLR model without adjustment for any confounders, only containing the exposure and the strata variable.

The estimated *CCS* effect in *CLogitTree* is slightly greater than the estimated effects from CLR and CLR^0, but the truth is unknown to us of course. The estimates for the exposure effect of *CLogitTree* with and without L_2 penalization hardly differ (1.835 vs 1.830). The confidence interval from *CLogitTree* is clearly wider than its counterparts, which is probably mostly explainable by the fact that this interval is determined via bootstrap instead of a closed formula.

7 | CONCLUDING REMARKS

The presented concept of CLR trees (*CLogitTree*) is an automated and data-driven machine learning approach for the analysis of matched case-control studies. It is an interesting alternative to conventional CLR with several advantages. First, it is more flexible as it uses less strict assumptions about the functional relationship between the covariates and the outcome variable. In particular, no assumption of linearity is needed and interactions between covariates are found and incorporated in a purely data-driven manner. In contrast, in CLR interactions have to be specified by the user in advance of the data analysis, which is challenging and therefore not done regularly in applications. Of course, in real word data also other kinds of association (like smooth non-linear effects) might appear. Yet, our tree-based approach offers a very flexible solution to many situations. Second, it is common practice in CLR to dichotomize ordinal or metric variables (eg, split age into young/old) in order to make them easier to handle and to avoid linearity assumptions. This is not necessary for *CLogitTree* as the relevant groups/partitions are detected automatically and purely data-driven when looking for the appropriate splits of the tree (yielding a much sparser predictor function). Third, the method performs automatic variable selection because not all potential covariates will be used for splits and are, therefore, removed from the final model.

We have shown, that the proposed method can compete with or outperform the standard method of CLR in situations where the DGP is non-linear. We do not claim the method to be superior to CLR in general, specifically in situations where the DGP is linear. We see *CLogitTree* as a flexible alternative to CLR, in the same way as classification and regression trees (CARTs) can be seen as a flexible alternative to linear or logistic regression.

This manuscript strongly focused on matched case-control studies. However, the proposed method is not restricted solely to this application but can be used as an alternative to CLR also in other circumstances. CLR can be used for the analyzes of clustered binary outcome variables as an alternative to mixed model approaches with random effects for the

clusters. For example, Graubard and Korn[28] proposed using CLR for the analysis of survey data. Accordingly, the method has a much wider range of applicability than the setting of matched case-control studies which was referred to in this manuscript.

All data analyzes in this manuscript have been done in R[29] with R version 4.1.1, the simulation study from Section 5 has been done with R version 4.0.3. The proposed method is implemented in R in the add-on package `CLogitTree` and publicly available from https://github.com/Schaubert/CLogitTree.

The package contains the method itself (including a pruning function for different values of the tuning parameter α and a function for pruning by BIC), a function to plot the resulting trees and a function to calculate bootstrap confidence intervals for the exposure effect. As common for tree methods, the user can further adapt the fitted trees via arguments for the maximal depth of the trees, the minimal node size in order to be eligible for further splitting and the minimum number of observations in any terminal node. Due to the internal use of permutation tests the method is computationally much more demanding compared to ordinary CLR. In particular, the determination of confidence intervals for the exposure effect via bootstrap is very time-consuming. For a considerable speed-up, the user has the option to run both the permutation tests and the bootstrap analysis parallelly on several cores. Currently, the method is not able to handle missing values and is restricted to a complete case analysis. However, the same holds for conventional routines for CLR.

ACKNOWLEDGEMENTS

The work of Moritz Berger was supported by the German Research Foundation (DFG), Grant BE 7543/1-1. Open Access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

The source code for the method introduced in this article is publicly available from Github and will be made available on CRAN in case the manuscript is accepted for publication. Data sharing is not applicable to this article as no new data were analyzed in this study.

ORCID

Gunther Schauberger https://orcid.org/0000-0002-0392-1580
Moritz Berger https://orcid.org/0000-0002-0656-5286

REFERENCES

1. Pratap A, Allred R, Duffy J, et al. Contemporary views of research participant willingness to participate and share digital data in biomedical research. *JAMA Netw Open*. 2019;2(11):e1915717. doi:10.1001/jamanetworkopen.2019.15717
2. Mansournia MA, Jewell NP, Greenland S. Case–control matching: effects, misconceptions, and recommendations. *Eur J Epidemiol*. 2018;33(1):5-14.
3. Lash TL, Vanderweele TJ, Haneuse S, Rothman KJ. *Modern Epidemiology*. 4th ed. Boca Raton: Wolters Kluwer; 2021.
4. Jewell NP. *Statistics for Epidemiology*. Boca Raton: CRC Press; 2003.
5. Pearce N. Analysis of matched case-control studies. *BMJ*. 2016;352:i969.
6. Breslow NE, Day NE, Halvorsen KT, Prentice RL, Sabai C. Estimation of multiple relative risk functions in matched case-control studies. *Am J Epidemiol*. 1978;108(4):299-307. doi:10.1093/oxfordjournals.aje.a112623
7. Kuo CL, Duan Y, Grady J. Unconditional or conditional logistic regression model for age-matched case-control data? *Front Public Health*. 2018;6:57.
8. Wan F, Colditz GA, Sutcliffe S. Matched versus unmatched analysis of matched case-control studies. *Am J Epidemiol*. 2021;190(9):1859-1866. doi:10.1093/aje/kwab056
9. Breslow NE, Day NE. *The Analysis of Case-Control Studies*. Statistical Methods in Cancer Research. Vol 1. Lyon: International Agency for Research on Cancer; 1980.
10. Hosmer DH, Lemeshow S. *Applied Logistic Regression*. New York: Wiley; 1989.
11. Avalos M, Pouyes H, Grandvalet Y, Orriols L, Lagarde E. Sparse conditional logistic regression for analyzing large-scale matched data from epidemiological studies: a simple algorithm. *BMC Bioinform*. 2015;16(S6):S1.
12. Reid S, Tibshirani R. Regularization paths for conditional logistic regression: the clogitl1 package. *J Stat Softw*. 2014;58(12):1-23.
13. Stanfill B, Reehl S, Bramer L, et al. Extending classification algorithms to case-control studies. *Biomed Eng Comput Biol*. 2019;10:1-12. doi:10.1177/1179597219858954
14. Zetterqvist J, Vermeulen K, Vansteelandt S, Sjölander A. Doubly robust conditional logistic regression. *Stat Med*. 2019;38(23):4749-4760. doi:10.1002/sim.8332
15. Breiman L, Friedman JH, Olshen RA, Stone JC. *Classification and Regression Trees*. Monterey, CA: Wadsworth; 1984.
16. Ripley BD. *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press; 1996.

17. Hothorn T, Hornik K, Zeileis A. Unbiased recursive partitioning: a conditional inference framework. *J Comput Graph Stat*. 2006;15:651-674.
18. Hothorn T, Lausen B. On the exact distribution of maximally selected rank statistics. *Comput Stat Data Anal*. 2003;43:121-137.
19. Shih YS. A note on split selection bias in classification trees. *Comput Stat Data Anal*. 2004;45:457-466.
20. Shih YS, Tsai H. Variable selection bias in regression trees with constant fits. *Comput Stat Data Anal*. 2004;45:595-607.
21. Strobl C, Boulesteix AL, Augustin T. Unbiased split selection for classification trees based on the gini index. *Comput Stat Data Anal*. 2007;52:483-501.
22. Puth MT, Tutz G, Heim N, Münster E, Schmid M, Berger M. Tree-based modeling of time-varying coefficients in discrete time-to-event models. *Lifetime Data Anal*. 2020;26:545-572.
23. Wright MN, König IR. Splitting on categorical predictors in random forests. *PeerJ*. 2019;7:e6339.
24. Su X, Tsai CL, Wang MC. Tree-structured model diagnostics for linear regression. *Mach Learn*. 2009;74(2):111-131.
25. Su X, Tsai CL. Tree-augmented Cox proportional hazards models. *Biostatistics*. 2005;6(3):486-499. doi:10.1093/biostatistics/kxi024
26. Zeileis A, Hothorn T, Hornik K. Model-based recursive partitioning. *J Comput Graph Stat*. 2008;17(2):492-514.
27. Tanaka LF, Schriefer D, Radde K, Schauberger G, Klug SJ. Impact of opportunistic screening on squamous cell and adenocarcinoma of the cervix in Germany: a population-based case-control study. *PLoS One*. 2021;16(7):1-17. doi:10.1371/journal.pone.0253801
28. Graubard BI, Korn EL. Conditional logistic regression with survey data. *Stat Biopharm Res*. 2011;3(2):398-408. doi:10.1198/sbr.2010.10002
29. R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing; 2021 http://www.R-project.org/

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Schauberger G, Tanaka LF, Berger M. A tree-based modeling approach for matched case-control studies. *Statistics in Medicine*. 2023;42(5):676-692. doi: 10.1002/sim.9637