Hindawi
Computational and Mathematical Methods in Medicine
Volume 2017, Article ID 7847531, 18 pages
https://doi.org/10.1155/2017/7847531

Research Article

Correcting Classifiers for Sample Selection Bias in Two-Phase Case-Control Studies

Norbert Krautenbacher,[1,2] Fabian J. Theis,[1,2] and Christiane Fuchs[1,2]

[1] *Institute of Computational Biology, Helmholtz Zentrum München, German Research Center for Environmental Health, Munich, Germany*
[2] *Department of Mathematics, Technische Universität München, Munich, Germany*

Correspondence should be addressed to Christiane Fuchs; christiane.fuchs@helmholtz-muenchen.de

Received 10 February 2017; Accepted 6 June 2017; Published 24 September 2017

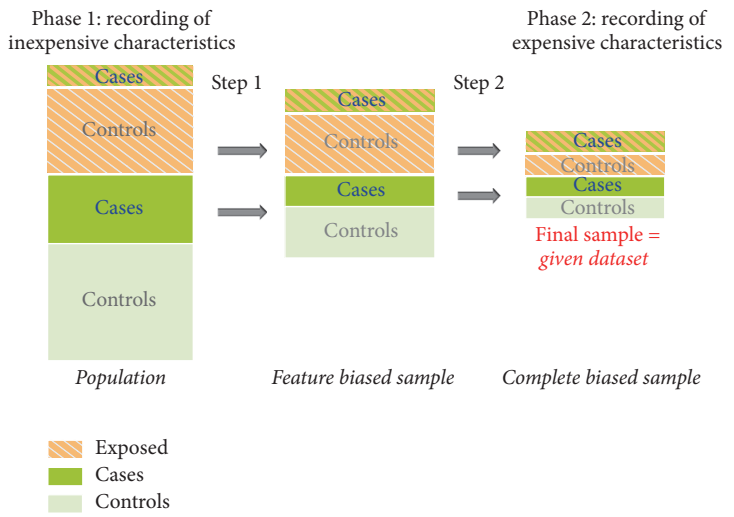
Academic Editor: Matthias Schmid

Epidemiological studies often utilize stratified data in which rare outcomes or exposures are artificially enriched. This design can increase precision in association tests but distorts predictions when applying classifiers on nonstratified data. Several methods correct for this so-called sample selection bias, but their performance remains unclear especially for machine learning classifiers. With an emphasis on two-phase case-control studies, we aim to assess which corrections to perform in which setting and to obtain methods suitable for machine learning techniques, especially the random forest. We propose two new resampling-based methods to resemble the original data and covariance structure: stochastic inverse-probability oversampling and parametric inverse-probability bagging. We compare all techniques for the random forest and other classifiers, both theoretically and on simulated and real data. Empirical results show that the random forest profits from only the parametric inverse-probability bagging proposed by us. For other classifiers, correction is mostly advantageous, and methods perform uniformly. We discuss consequences of inappropriate distribution assumptions and reason for different behaviors between the random forest and other classifiers. In conclusion, we provide guidance for choosing correction methods when training classifiers on biased samples. For random forests, our method outperforms state-of-the-art procedures if distribution assumptions are roughly fulfilled. We provide our implementation in the R package *sambia*.

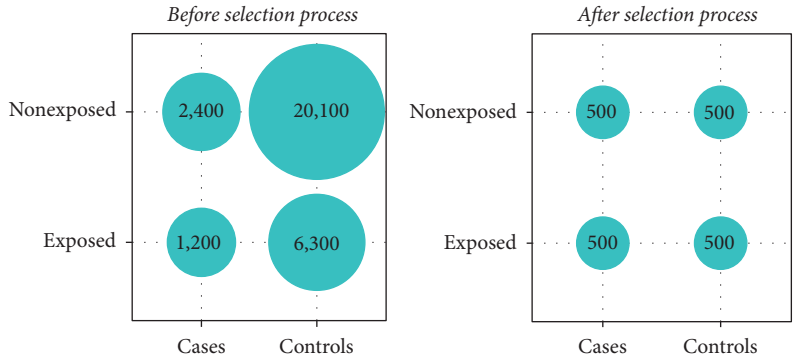
1. Introduction

Statistics is an art of inferring information about large populations from comparably small random samples. This is necessary because in practice it is most often impossible to receive measurements from all individuals in a population (e.g., due to organizational or cost reasons). In the clinical context, for example, one might aim to predict the risk for a certain disease based on clinical features for an entire population. The risk model will be derived from information from a much smaller random subsample of the population. When building such models, a common assumption is that the subsample follows the same distribution as the population the sample was taken from. This assumption, however, is not valid if the sample is not taken at random. In the epidemiological context, for example, this case occurs in the well-known *case-control studies* [1]. Here, one is interested in finding associations between features and rare disease outcomes. In order to increase precision and achieve higher statistical power for finding significant associations, cases are enriched such that cases and controls are equally represented in the sample. When a case-control study is used for risk prediction on an unbiased population (e.g., via logistic regression), certain adjustments have to be made which have been elaborated in [2–5].

An even more complex sample design appears in *two-phase case-control studies* [6, 7]. Here, one enriches not only a rare disease outcome but also a rare covariate (e.g., an exposure). This measure prevents the sample from containing only few individuals that fall into both rare categories. From such a sample, one would hardly be able to draw conclusions about the rare combination. Figure 1(a) illustrates how the sampling procedure is performed in practice. Figure 1(b) shows an exemplary table of numbers of cases/controls and



(a) Stratified random selection process of a two-phase case-control study. Feature characteristics known about a whole finite population are typically features which are inexpensive to measure and called characteristics recorded in Phase 1. The expensive characteristics are recorded only in Phase 2—in the final sample



(b) Exemplary cross table for data before (left) and after (right) the selection process of a two-phase case-control study. There is a clear dependency between exposure and disease in the population. After the sampling process, this dependency vanishes completely for the final sample

Figure 1

exposed/nonexposed individuals in the population and the sample. This and other complex survey designs (e.g., cohort sampling designs [8]) have been used in order to obtain subpopulations with rare characteristics of features of interest [9–11]. The efficiency and analysis of the design are described in [6].

In the situations described above, the sample follows a different distribution than the population. This can affect statistical analysis. In the general context, this issue is known as *sample selection bias* [12–14]. It generally occurs when not all individuals from the population have the same probability of getting selected for the sample. If a statistical estimate is affected by sample selection bias, one should correct for it. The question of whether correction is necessary depends on the type of sample selection bias, the considered classifier, and the research question to be answered. For example, no adjustment is required if only the outcome variable is enriched and logistic regression is applied for prediction purposes, because the slope coefficients of the linear predictor remain asymptotically unaffected by sample selection bias for this case (if the functional form and the explanatory features for the model are correct) [15]. In general, however, correction is required, and there are several solutions to encounter this problem in complex survey designs [16, 17]. These existing approaches mainly focus on classical prediction methods or simple survey designs. Strategies applicable also for machine learning approaches have been suggested in the general sample selection bias context [12, 18, 19]. These methods reconstruct the population data or its covariance structure and typically involve nonparametric resampling techniques like bootstrapping. However, they neglect complex survey designs. Thus, while correcting for sample selection bias in logistic regression is well investigated, its consideration is unclear for most machine learning approaches.

This paper assesses, proposes, and compares approaches to correct for sample selection bias in complex surveys,

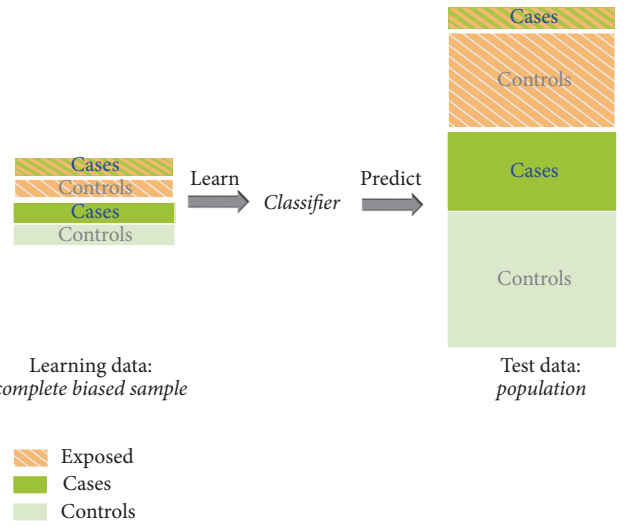


FIGURE 2: Scheme of learning on biased learning data and predicting on unbiased test data. The classifier learns on four equally sized strata (complete biased learning data set) but predicts on a data set (unbiased population) of different sizes of the four strata.

especially in two-phase case-control studies. Therefore, we focus on the binary outcome. Figure 2 illustrates the issue to be addressed. The emphasis is on a widely used machine learning approach: the random forest. We correct for the covariance structure of the sample by incorporating knowledge about the sample selection procedure into nonparametric and parametric resampling techniques. As the random forest is based on resampling anyway (in terms of bagging; see Section 3.2), we incorporate the correction step into the inherent resampling procedure. We compare our correction approaches to analogous state-of-the-art approaches, both for the random forest and for other common classifiers, namely, logistic regression, logistic regression including interaction terms, and the naive Bayes classifier. We especially address the question of whether correction is necessary in random forests, and if so, whether current correction approaches can successfully be transferred to the random forest and whether improvement is possible through alternative approaches. We assess and compare the prediction performance of the correction techniques in a synthetic simulation study and in a real data application. We provide the R package *sambia* so that readers can easily apply the methods presented here to their data.

This paper is structured as follows. We formalize sample selection bias and address the necessity of correction in Section 2. Section 3 explains current approaches for corrected learning on biased samples, and we propose two new methods based on drawing observations from theoretical distributions assumed for the given data. We furthermore analyze properties of the various approaches in the context of sample selection bias. Section 4 presents a simulation study which compares all approaches regarding performance on new unbiased test data. Section 5 shows a similar analysis on real data. We discuss and conclude our work in Section 6.

2. Preliminaries

This section introduces general definitions and background information: a formal description of sample selection bias (Section 2.1), the special case of two-phase case-control studies (Section 2.2), and properties of biased samples (Section 2.3).

2.1. Sample Selection Bias. The following setup is similar to that of Zadrozny [12] and distinguishes *sample selection bias* into three types. We assume a set of observations $\{(\mathbf{x}_i, y_i)\}_{i=1,\dots,n}$ which are drawn independently from a distribution D. The domain of D is $\mathcal{X} \times \mathcal{Y}$ with $\mathcal{X}$ being the feature space and $\mathcal{Y}$ being a measurable space. Here, $\mathcal{Y}$ is a discrete binary label space since we focus on binary classifiers in this work. Throughout the paper, we will denote random variables by capital letters and realizations (i.e., observations in the sample) by lowercased letters.

For the setup of the sample selection bias issue, let in addition $\mathcal{S}$ be a binary space. $S \in \mathcal{S}$ is the variable that controls the selection of observations: For $s_i = 1$, the ith observation is selected; for $s_i = 0$, the observation is not selected. Thus, observations $(\mathbf{x}_i, y_i, s_i)$ are drawn from a distribution $\mathcal{L}$ with domain $\mathcal{X} \times \mathcal{Y} \times \mathcal{S}$.

In general, a sample $\{(\mathbf{x}_i, y_i, s_i)\}_{i=1,\dots,n}$ can be biased in three different ways. These types of sample selection bias can be described as follows [12, 19]:

(i) Label bias: biasedness depends on Y only, so $P(S \mid \mathbf{X}, Y) = P(S \mid Y)$ but $P(S \mid Y) \neq P(S)$.

(ii) Feature bias: biasedness depends on $\mathbf{X}$ only, so $P(S \mid \mathbf{X}, Y) = P(S \mid \mathbf{X})$ but $P(S \mid \mathbf{X}) \neq P(S)$.

(iii) Complete bias: biasedness depends on $\mathbf{X}$ and Y; that is, there is no independence between S and $\mathbf{X}, Y$, so $P(S \mid \mathbf{X}, Y) \neq P(S \mid Y)$ and $P(S \mid \mathbf{X}, Y) \neq P(S \mid \mathbf{X})$.

Under label bias, S is not necessarily independent of $\mathbf{X}$ ([19]; for details, see also Appendix A), and for feature bias S is not necessarily independent of Y.

Whenever there is sample selection bias, there are *selection probabilities* $P(S = 1 \mid Y, \mathbf{X})$ (in particular $P(S = 1 \mid Y)$ for label bias and $P(S = 1 \mid \mathbf{X})$ for feature bias). In practice, these probabilities can often be estimated if they are unknown. Throughout this paper, we assume them to be provided. All approaches proposed in this paper will incorporate these selection probabilities in terms of weights corresponding to the inverse probabilities $P(S = 1 \mid \mathbf{X}, Y)^{-1}$.

2.2. Sample Selection Bias in Two-Phase Case-Control Studies. In this paper, we will discuss the special case of two-phase case-control studies and hence put them into the context of sample selection bias in this subsection.

The case-control study is an example for sample selection bias in the clinical context: Some diseases under investigation are very rare in the entire population. A random sample of study participants would contain very few cases of the disease. Statistical analysis would suffer from low precision and thus low power. In order to increase precision and power, the number of cases is enriched such that the proportion of cases

and controls in the sample is identical. In particular, $P(Y = 1 \mid S) = 0.5$ whereas the prevalence rate $P(Y = 1)$ is much smaller, so $P(Y = 1 \mid S) \neq P(Y = 1)$. This by Bayes' theorem implies $P(S \mid Y = 1) \neq P(S)$, and thus there is label bias.

Case-control studies are mostly used for investigating associations between disease and features. The underlying label bias does not alter the effect estimates in hypothesis testing for associations between disease and features. However, this is true only asymptotically, and there may be consequences in small sample scenarios. If one focuses on prediction, for example, via logistic regression, as we do in this paper, the intercept estimate can simply be adjusted as described in Rose and van der Laan [4] or Steyerberg et al. [2]. Elkan [20] offers a solution for arbitrary classifiers.

In *two-phase case-control studies*, on the other hand, the selection is additionally controlled by a categorical feature variable. Such studies suffer from label *and* feature bias, so there is complete bias. We focus on this case (i.e., complex survey designs which involve complete bias).

2.3. Stratified Random Samples. When data is sampled as in one-phase or two-phase case-control studies, there are groups within which the selection probabilities are equal. These groups are called *strata*. In this paper, we focus on two-phase case-control studies where the strata are determined by a categorical stratum feature (often an exposure) X_e and the outcome Y. The remaining features of $\mathbf{X}$ are $\widetilde{\mathbf{X}} := \mathbf{X} \setminus X_e$.

For a population of size N and sample size n, let $h \in \{1, \ldots, H\}$ be the index of the stratum. Realizations falling into stratum h are denoted by $\tilde{\mathbf{x}}_h$, x_{eh}, and y_h or combined as $(\mathbf{x}_h, y_h) = (\tilde{\mathbf{x}}_h, x_{eh}, y_h)$. We denote by n_h the size of the stratum h in the sample and by N_h its size in the population. Then, clearly, $P(S = 1) = n/N$ and

$$P(S = 1 \mid \mathbf{x}, y) = P(S = 1 \mid x_e, y) = P(S = 1 \mid h(x_e, y)) = \frac{n_{h(x_e,y)}}{N_{h(x_e,y)}}, \quad (1)$$

where $h(x_e, y)$ denotes the stratum determined by x_e and y. Throughout the paper, we will simply abbreviate this by h.

If the features determining the selection probabilities are categorical, the data set can be partitioned into corresponding strata with equal selection probabilities. This is not the case if, for example, the feature causing the selection bias is continuous. In the categorical case, selection probabilities can be used for adjusting the distribution of the sample to the original distribution of the population.

Consider the selection probability $P(S = 1 \mid h)$ for an observation of stratum h. We define

$$w_h := \left[\frac{\max_{h'} P(S = 1 \mid h')}{P(S = 1 \mid h)} \right] \quad (2)$$

as the *inverse-probability (IP) weight* for stratum h. The squared brackets denote rounding to the closest integer. The term *IP weight* is sometimes used in the literature for the simple inverse selection probability $P(S = 1 \mid h)^{-1}$. In this work, we use w_h rather than $P(S = 1 \mid h)^{-1}$ to keep the number of newly generated observations minimal.

In our correction approaches, we will use

$$n' := \sum_{h=1}^{H} n_h w_h, \quad (3)$$

which can be seen as the number of reweighted observations (i.e., the sum of all observations multiplied by their weights). As stated above, we are interested in adjustment methods which can be applied to arbitrary classifiers. In the next section, after stating a typical setup of a statistical learning procedure, we will describe several sample selection bias correction approaches proposed in the literature.

3. Methods

In this section, we describe, modify, and analyze IP weight-incorporating classifiers which are designed for learning on an unbiased data set, when only a biased data set for learning is given.

3.1. Correction Approaches. All approaches adjust the given data set to correct for sample selection bias by reconstructing the original (unbiased) data structure before or while learning the classifier. We consider the classifier

$$\varphi : \begin{cases} (\mathcal{X} \times \mathcal{Y})^{\times n} \times \mathcal{X} \longrightarrow \mathcal{Y} \\ ((\mathbf{x}, y), \mathbf{X}) \longmapsto \varphi((\mathbf{x}, y); \mathbf{X}), \end{cases} \quad (4)$$

where the given learning data set $(\mathbf{x}, y) = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))$ is mapped to the prediction (in our case classification) rule and applied to the random variable $\mathbf{X}$.

3.1.1. State of the Art. The methods in this section were proposed in the literature and are partly modified for our purposes.

No Correction. The naive approach for learning on a biased sample is to simply ignore the bias. No IP weights are used, and the classifier is trained on the given sample as it is. As shown by Zadrozny [12], this approach is valid for some cases of sample selection bias, namely, for feature bias for a specific type of classifiers.

Inverse-Probability Oversampling. An intuitive method for correcting for sample selection bias is the plain replication of each observation in the sample according to its IP weight (i.e., in a stratified random sample, one replicates an observation of stratum h by the factor w_h). Then, the number of observations in the reconstructed sample is n'. This sample is used for learning. In maximum likelihood-based approaches like generalized regression models, this method is equal to weighting the single likelihoods per observation. The procedure, sometimes simply called *inverse-probability weighting*, has been used early [21], with applications both in regression [22] and in general statistical learning [20]. We refer to this technique as *IP oversampling*: Since in the stratification process some observations were *oversampled*, this method is a way of reoversampling underrepresented observations

in the stratified sample. Since IP oversampling is applicable to arbitrary classifiers, we take it into account for further comparisons. A drawback is that it changes the covariance structure per stratum h. In Section 3.1.2, we propose a method that corrects for this issue.

Inverse-Probability Bagging. Another correction method uses bootstrap aggregation and averaging, commonly abbreviated to the acronym *bagging*. The procedure averages several predictions trained on an ensemble of bootstrap samples and thus makes learners more robust [23]. Nonparametric bootstrap samples arise by randomly drawing n times from the original data set of size n with replacement. Bagging procedures fit a learner on each of these bootstrap samples and combine the learners by averaging predictions or by majority vote. When building bootstrap samples from biased data sets, as in our case, resampling can take into account IP weights: Instead of drawing observations randomly, selection probabilities are set proportional to w_h for the respective strata h. This procedure is proposed by Nahorniak et al. [24] and labeled as *IP bagging* here.

Costing. Zadrozny et al. [18] argue that sampling with replacement as done in IP bagging is inappropriate since sets of independent observations from continuous distributions contain two identical elements only with zero probability, whereas nonparametric bootstrap samples generally contain observations repeatedly. Zadrozny et al. [18] propose an approach called *costing*, which is similar to IP bagging in terms of resampling from the learning data and aggregation of learned algorithms on m new samples. It differs in the implementation of resampling the m learning sets: Here, an observation from the original learning set enters a resampled data set only once at most. It is selected with probability $w_h/\max_{h'} w_{h'}$ according to the corresponding stratum h. Consequently, the size of the new samples is smaller than n and generally varies among the m learning sets. The latter aspect indicates the difference of this approach to subsampling without replacement. A detailed description of the aspects of the algorithm can be found in Zadrozny et al. [18], Sections 2.3.2 to 2.3.4.

A drawback of costing in case of strata with a low number of observations is the following: There may be subsamples which do not contain observations from all strata, which implies that no classification rule can be learnt for the missing strata from those subsamples. For the purposes of this paper, we adjusted the costing algorithm by not taking into account such incomplete samples. This modification causes bias which we consider negligible.

Modified SMOTE. So far, all correction approaches replicated given observations. In contrast, [25] proposed a *synthetic minority oversampling technique (SMOTE)* to generate new, synthetic data. The strategy is designed as a solution for the imbalanced class problem, where rare cases (the *minority class*) are hardly represented in the (nonstratified) sample, which mainly consist of common cases from the *majority class*. In this situation, several classifiers perform poorly because of the imbalanced proportion of outcome categories in the data.

In its original form, SMOTE generates synthetic observations for the minority class as follows: For fixed $k \in \mathbb{N}$, one determines the k nearest neighbors of the minority class. Depending on the desired number of new observations, one then randomly selects a corresponding amount of instances from this neighborhood. New observations arise as weighted averages between original feature vectors and selected nearest neighbors. To that end, weights are randomly sampled from the unit interval.

We adapt SMOTE to the context of stratified random samples: Rather than enlarging only the minority class, we generate synthetic observations for all strata with $w_h > 1$. Thus, we apply SMOTE up to $H - 1$ times, once for each stratum which requires more observations. We refer to this algorithm as *modified SMOTE* hereafter.

3.1.2. Correcting Covariance Structures. The approaches above aim to reconstruct the original data distribution in order to then learn a classifier on an unbiased sample. However, several aspects are not incorporated so far: IP oversampling replicates observations and by this biases the covariance structure within the strata. A correction for this biasedness should be provided. Similarly, modified SMOTE biases the data, especially for large weights w_h, where the same observations are used several times for synthetic data generation and lack contributing sufficient variation. IP bagging and costing are both exclusively based on resampling observed data. This may become problematic especially for small sample sizes or only small stratum sizes (which can occur in the resampled data sets for these two approaches): The fine structure in the given data can be spurious due to the deficit of observations. Also, due to small sample sizes and hence too few values in the sample only covering a restricted range, one may underestimate variance and covariance of the data.

In this section, we propose two procedures which aim to conquer the problem of small strata by increasing the number of observations per stratum and at the same time estimate the covariance of the population appropriately. The idea behind both approaches is to exploit the fact that within each stratum h all observations are assigned the same weight w_h. This enables parametric resampling within each stratum.

Let $\widetilde{\mathscr{L}}_h$ be the distribution which $\widetilde{\mathbf{X}}_h$ follows. We aim to approximate $\widetilde{\mathscr{L}}_h$ by theoretical distributions and estimate their parameters for each stratum h. In practice, determining the multivariate distribution of the features is difficult and relies on assumptions. One might, for example, assume normally distributed features,

$$\widetilde{\mathbf{X}}_h \sim \mathcal{N}\left(\mu_h, \Sigma_h\right), \tag{5}$$

and would then have to estimate $\widehat{\mu}_h$ and $\widehat{\Sigma}_h$ for all h, which is typically done by their empirical pendants. Even though we focus on the normal distribution in our empirical investigations, we propose the following approaches such that they can be applied to arbitrary distribution assumptions.

Stochastic Inverse-Probability Oversampling. Our first approach builds upon the re- or oversampling techniques

Table 1: Properties and performance of correction approaches for logistic regression and random forest. The properties are as follows: (i) a correction attempt is made at all; (ii) the covariance structure of the learning data is attempted to be unbiased; (iii) learning is based on a data set containing a larger number n' of observations than the original stratified data set (see (3)). Criteria are fulfilled ("✓"), not clearly fulfilled ("(✓)"), or not fulfilled ("×").

Correction approach	Properties according to Section 3.1.3			Sufficient performance	
	(i)	(ii)	(iii)	Logistic regression	Random forest
No correction	×	×	×	×	×
IP oversampling	✓	×	✓	✓	×
IP bagging	✓	✓	×	✓	×
Costing	✓	✓	×	(✓)	×
Modified SMOTE	✓	(✓)	✓	(✓)	×
Stochastic IP oversampling	✓	✓	✓	✓	×
Parametric IP bagging	✓	✓	✓	✓	✓

described in Section 3.1.1. However, the repeated occurrence of observations of continuous features falsifies the covariance structure of the reconstructed samples. Hence, we add noise to those data sets obtained via IP oversampling and thus call our proceeding *stochastic IP oversampling*.

When adding this noise, we want to retain important distribution characteristics of the respective stratum. As stated above, the stratified sample contains features $\widetilde{\mathbf{X}}_h \sim \widetilde{\mathscr{L}}_h$. After performing IP oversampling, the reconstructed features $\widetilde{\mathbf{X}}'_h$ do not follow $\widetilde{\mathscr{L}}_h$ anymore. We aim to adjust $\widetilde{\mathbf{X}}'_h$ by adding noise terms $\widetilde{\boldsymbol{\varepsilon}}_h$ such that $\widetilde{\mathbf{X}}'_h + \widetilde{\boldsymbol{\varepsilon}}_h$ approximately follows the original distribution $\widetilde{\mathscr{L}}_h$ in the sense that it agrees in expectation and covariance. In the following, we derive a respective distribution $\widetilde{\mathscr{L}}_h^{\text{adj}}$ for $\widetilde{\boldsymbol{\varepsilon}}_h$.

We seek two conditions to hold:

$$\mathbb{E}\left(\widetilde{\mathbf{X}}'_h + \widetilde{\boldsymbol{\varepsilon}}_h\right) = \mathbb{E}\left(\widetilde{\mathbf{X}}_h\right), \tag{6}$$

$$\operatorname{cov}\left(\widetilde{X}_h^{(k)\prime} + \widetilde{\varepsilon}_h^{(k)}, \widetilde{X}_h^{(j)\prime} + \widetilde{\varepsilon}_h^{(j)}\right) = \operatorname{cov}\left(\widetilde{X}_h^{(k)}, \widetilde{X}_h^{(j)}\right) = \Sigma_h \tag{7}$$

for all $k, j \in \{1, \ldots, p\}$ denoting the index of the features. Because of (6) and since $\mathbb{E}(\widetilde{\mathbf{X}}'_h) = \mathbb{E}(\widetilde{\mathbf{X}}_h)$, we obtain

$$\mathbb{E}\left(\widetilde{\boldsymbol{\varepsilon}}_h\right) = 0. \tag{8}$$

In the Appendix (Appendix A, (B.3)), we derive the adjusted *noise covariance matrix* $\Sigma_h^{\text{adj}} := \operatorname{cov}(\widetilde{\varepsilon}_h^{(k)}, \widetilde{\varepsilon}_h^{(j)})$, which leads to

$$\Sigma_h^{\text{adj}} = \frac{w_h - 1}{w_h n_h - 1} \Sigma_h. \tag{9}$$

For instance, when assuming a multivariate normal distribution $\widetilde{\mathbf{X}}_h \sim \widetilde{\mathscr{L}}_h = \mathcal{N}(\mu_h, \Sigma_h)$, the noise term

$$\widetilde{\boldsymbol{\varepsilon}}_h \sim \widetilde{\mathscr{L}}_h^{\text{adj}} = \mathcal{N}\left(0, \frac{w_h - 1}{w_h n_h - 1} \Sigma_h\right) \tag{10}$$

would retain the stratum expectation and covariance (and thus in the Gaussian case the entire distribution).

In order to make a corresponding correction method more robust, we repeat the noise-adding procedure and average over the models fitted on each of those repetitions. Algorithm 1 displays the single steps of stochastic IP oversampling.

Parametric Inverse-Probability Bagging. Stochastic IP oversampling above consisted of a deterministic replication of observations followed by a stochastic alteration by adding noise. Now, we propose a completely parametric approach which we call *parametric IP bagging*. As in IP bagging, we draw bootstrap samples from the original stratified data set. This time, however, we employ parametric instead of nonparametric bootstrap and set the bootstrap sample size to n'. As in stochastic IP oversampling, we assume a multivariate distribution underlying the original data and estimate the parameters stratum-wise. The procedure is defined by Algorithm 2.

3.1.3. Properties of Correction Approaches. So far, we described seven ways to deal with sample selection bias: no correction, IP oversampling, IP bagging, costing, modified SMOTE, stochastic IP oversampling, and parametric IP bagging. This subsection compares their characteristics. They are summarized in the left part of Table 1.

(i) Incorporation of Weights. Except for the noncorrection approach, all correction methods incorporate weights. As mentioned in Section 3.1.1, there are cases of sample selection bias where the bias does not affect the classifier so that correction in terms of weighting is not necessary. However, as we will elaborate in this paper on two-phase case-control studies, correction is necessary in the context of complete bias.

(ii) Correcting Covariance Structure of Learning Data. Sample selection bias can cause a biased covariance structure in the data. Some but not all correction approaches correct for this bias: The noncorrection approach clearly uses the biased covariance structure. Also, IP oversampling does not correct for it; the replication of observations generally leads to underestimating the covariance (cf. (B.2) in the Appendix). For modified SMOTE, the resulting covariance structure depends on the magnitude of the weights w_h and the degree of separation of the features into distinct clusters.

Input: Observed sample $(\tilde{\mathbf{x}}, x_e, y)$ of size n, IP weights w_h
Output: Unbiased prediction $\hat{y}$ for new unbiased data $(\mathbf{X}, Y) \sim D$
(1) Perform IP oversampling, resulting in reconstructed sample $(\tilde{\mathbf{x}}', x_e', y')$ of size n'
(2) **for** $b = 1$ *to* B **do**
  for $h = 1$ *to* H **do**
    (a) Estimate Σ_h^{adj} of distribution $\widetilde{\mathscr{L}}_h$
    (b) Draw noise vector $\tilde{\boldsymbol{\varepsilon}}_h^b$ from $\widetilde{\widetilde{\mathscr{L}}}_h^{\text{adj}}$ of length $n_h w_h$
    (c) Rebuild original stratum as $(\tilde{\mathbf{x}}_h' + \tilde{\boldsymbol{\varepsilon}}_h^b, x_{eh}', y_h')$
  end
    (a) Combine strata to sample:
      $(\tilde{\mathbf{x}}' + \tilde{\boldsymbol{\varepsilon}}^b, x_e', y') = ((\tilde{\mathbf{x}}_1' + \tilde{\boldsymbol{\varepsilon}}_1^b, x_{e1}', y_1'), \ldots, (\tilde{\mathbf{x}}_H' + \tilde{\boldsymbol{\varepsilon}}_H^b, x_{eH}', y_H'))$
    (b) Fit classifier $\hat{y}^b = \varphi((\tilde{\mathbf{x}}' + \tilde{\boldsymbol{\varepsilon}}^b, x_e', y'); \mathbf{X})$
  end
(3) Output the ensemble of learners $\{\hat{y}^b\}_{b=1,\ldots,B}$
(4) Aggregate predictions on new data set by averaging: $\hat{y} = \sum_{b=1}^{B} \hat{y}^b$

Algorithm 1: Stochastic inverse-probability oversampling.

Input: Observed sample $(\tilde{\mathbf{x}}, x_e, y)$ of size n, IP weights w_h
Output: Unbiased prediction $\hat{y}$ for new unbiased data $(\mathbf{X}, Y) \sim D$
(1) **for** $b = 1$ *to* B **do**
  for $h = 1$ *to* H **do**
    (a) Estimate parameters of distribution $\widetilde{\mathscr{L}}_h$
    (b) Draw parametric bootstrap sample $\tilde{\mathbf{x}}_h^b$ from $\widetilde{\widetilde{\mathscr{L}}}_h$ of size $n_h w_h$
    (c) Rebuild stratum as $(\tilde{\mathbf{x}}_h^b, x_{eh}^{\times w_h}, y_h^{\times w_h})$, where "$\times w_h$" denotes w_h-fold concatenation
  end
    (a) Combine strata to sample:
      $(\tilde{\mathbf{x}}^b, x_e^{\times w}, y^{\times w}) = ((\tilde{\mathbf{x}}_1^b, x_{e1}^{\times w_1}, y_1^{\times w_1}), \ldots, (\tilde{\mathbf{x}}_H^b, x_{eH}^{\times w_H}, y_H^{\times w_H}))$
      with $w = \sum_{h=1}^{H} w_h$
    (b) Fit classifier $\hat{y}^b = \varphi((\tilde{\mathbf{x}}^b, x_e^{\times w}, y^{\times w}); \mathbf{X})$
  end
(2) Output the ensemble of learners $\{\hat{y}^b\}_{b=1,\ldots,B}$
(3) Aggregate predictions on new data set by averaging: $\hat{y} = \sum_{b=1}^{B} \hat{y}^b$

Algorithm 2: Parametric inverse-probability bagging.

For instance, a stratum with large weight w_h will cause a large number of newly generated observations as compared to the original number of observations. The same neighbors will be selected several times such that sufficient variation of the new observations cannot be guaranteed. This may result in a similar issue as for IP oversampling described above. All other approaches aim to obtain the right covariance structure per stratum and in the entire reconstructed sample.

(iii) Size of Reconstructed Samples. As a well-known fact in statistical learning, the bias of a classifier increases when the learning sample size decreases. IP bagging is based on reconstructed samples of the same size n as the original stratified data set. Sample sizes in costing are even smaller and vary between bootstrap samples. Particularly, the small strata contain a small number of observations for these two ways of reconstructing the sample. Consequently, a certain structure of the data may get lost for learning (e.g., the appropriate variability within small strata may not be given anymore). IP oversampling, modified SMOTE, and our own methods, stochastic IP oversampling and parametric IP bagging, on the other hand, employ reconstructed samples of larger sizes n' as defined in (3). By this, we intend to have sufficient numbers of observations in each stratum for possibly improving the learning of the classifier as compared to the use of smaller samples. In the nonparametric IP oversampling, the larger sample size induces a large number of perfectly repeated observations. This, again, biases the covariance structure. In our parametric approaches, stochastic IP oversampling and parametric IP bagging, this drawback does not occur.

3.2. Classifiers. In Sections 3.1.1 and 3.1.2, several approaches adjusting for sample selection bias have been presented and proposed. We implemented all approaches for the following classifiers: classical logistic regression based on maximum likelihood estimation as a classifier serving as reference since correction approaches are well established for it, the tree-based random forest as our main object of interest, and

logistic regression including interaction terms and the naive Bayes classifier as further algorithms for comparison.

As described by Zadrozny [12], a classifiers' output can depend either on $P(Y \mid \mathbf{x})$ only or on both $P(Y \mid \mathbf{x})$ and $P(\mathbf{X})$. The first type of classifiers per definition is not affected by feature bias whereas the second type is affected. Thus, one has to consider that the two types behave differently under complete bias, as well.

Logistic Regression. We employ logistic regression [26] as a common classical binary classification method. The model assumes $Y \mid \mathbf{X}$ to be Bernoulli distributed with success probability

$$P(Y = 1 \mid \mathbf{X}) = \left(1 + \exp\left\{-\left(\beta_0 + \mathbf{X}\boldsymbol{\beta}\right)\right\}\right)^{-1}, \quad (11)$$

where β_0 and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ are unknown parameters representing the effects of the features $\mathbf{X}$ on the outcome variable Y.

We investigate two variants of this model: Once, all features enter the model just linearly. In a refinement, features are additionally included as all possible two-way interaction term combinations, not only in order to detect possible interaction effects but also to obtain more complex decision boundaries.

Random Forest. Random forests are ensembles of decision trees and a modification of bagging [27]. The basic procedure of the learning algorithm is the following:

(1) A bootstrap sample is drawn from the given learning data set.

(2) A decision tree is grown by constructing recursive binary splits to the given data based on the features.

(3) At each node only a subset of features is selected at random.

(4) Steps (1) to (3) are repeated and all trees are averaged; class probabilities can be estimated as the relative frequency of the class of interest for a terminal node.

An essential step which is different from common bagging (cf. Section 3.1.1) is Step (3). The random selection of features decorrelates the trees and makes the bagging procedure more efficient. For all approaches in Sections 3.1.1 and 3.1.2 which are based on aggregating after resampling, namely, IP bagging, costing, stochastic IP oversampling, and parametric IP bagging, we incorporate these approaches into the random forest correspondingly. That means, instead of performing bagging within another bagging, we combine the two procedures. Note that IP oversampling incorporated in a random forest turns the approach to a bagging method. In fact, IP oversampling is exactly the same method as IP bagging when using samples of size n' instead of n. Thus, for the implementation of our approaches into the random forest, we implicitly take both versions of IP bagging into account.

Naive Bayes. The naive Bayes classifier is another common machine learning algorithm for classification (see, e.g., Hastie et al. [28]). It assumes independence between the p features and simply calculates for each class j that can be attained by Y the marginal classifier

$$\varphi^{(j)}(\mathbf{X}) = \prod_{k=1}^{p} \varphi^{(j,k)}\left(\mathbf{X}^{(k)}\right) \quad (12)$$

by estimating feature-wise classifiers $\varphi^{(j,k)}$ via one-dimensional kernel-density estimation. That means the impact of each feature $\mathbf{X}^{(k)}$ is estimated separately and combined to an overall classifier.

4. Simulation Study

So far, we have presented and developed strategies for fitting classifiers under complete bias. In this section, we investigate their performance when a sample from a two-phase case-control study is given as learning data set but the test data is unbiased (i.e., it is a random sample from the population). We do this in a simulation study. After stating the setup in Section 4.1, we compare performances for the introduced correction approaches (Section 3.1) and classifiers (Section 3.2) and report the results in Section 4.2.

4.1. Design. For evaluating the performance of correction approaches on training samples from two-phase case-control studies and unbiased validation data sets, we need three kinds of data sets: first, a biased learning data set stemming from a two-phase case-control study; second, an unbiased large reference learning data set for comparison purposes (we refer to this data as *population*; it is not available in practice); third, an unbiased test data set distributed like the population. We artificially simulated such data sets as described in the following.

We started by generating the large unbiased population data set. To that end, we randomly sampled 10^5 feature vectors consisting of one binary exposure variable X_e and $p = 5$ continuous other features $\widetilde{X}^{(j)}$, $j \in \{1, \ldots, 5\}$. The exposure X_e was meant to serve as a stratum feature with a low proportion (10%) of exposed ($X_e = 1$) individuals and a majority of nonexposed ($X_e = 0$) individuals. The $p = 5$ other features were generated independently of x_e and of each other. We investigated the following four distribution families:

(i) Normal distribution: $\widetilde{X}^{(j)} \sim \mathcal{N}(\mu^{(j)}, \sigma^{(j)^2})$ for all $j = 1, \ldots, p$

(ii) Student's t-distribution: $\widetilde{X}^{(j)} \sim \mathrm{t}(\nu_j)$ for all $j = 1, \ldots, p$

(iii) Poisson distribution: $\widetilde{X}^{(j)} \sim \mathrm{Po}(\lambda_j)$ for all $j = 1, \ldots, p$

(iv) Bernoulli distribution: $\widetilde{X}^{(j)} \sim \mathrm{Ber}(\pi_j)$ for all $j = 1, \ldots, p$

The distribution parameters were uniformly drawn from the following sets for $j = 1, \ldots, p$: $\mu^{(j)} \in [1, 10]$, $\sigma^{(j)} \in [1, 5]$, $\nu_j \in \{10, 11, 12, \ldots, 98, 99, 100\}$, $\lambda_j \in \{1, 2, 3, 4, 5\}$, and $\pi_j \in [0.4, 0.6]$.

In order to also investigate more realistic distribution scenarios, we additionally generated and analyzed data sets with

dependent features and features from different distributions. These studies yield similar results as the setting above and are described in the Supplementary Material of this paper (available online at https://doi.org/10.1155/2017/7847531).

Given the covariates $\mathbf{X} = (X_e, \widetilde{\mathbf{X}})$, the outcome Y was generated according to a logistic regression model: $Y \mid \mathbf{X} \sim \text{Ber}(\theta(\mathbf{X}))$, where $\theta(\mathbf{X}) = (1 + \exp\{-(\beta_0 + \mathbf{X}\boldsymbol{\beta})\})^{-1}$. We chose the effects in terms of regression coefficients $\boldsymbol{\beta} = (\beta_e, \beta_1, \ldots, \beta_5)'$ as follows: The exposure has a negative effect on the outcome with $\beta_e := \log 0.5$. The effects $\beta_1, \ldots, \beta_5$ for the main features are varied at random, namely, uniformly on the interval $[-0.15, 0.15]$ in order to gain an intermediate performance of a classifier applied on an independent data set. β_0 was chosen such that $P(Y = 1) = 0.1$. By this setup, the population with a rare exposure, $P(X_e = 1) = 0.1$, and rare cases, $P(Y = 1) = 0.1$, is fully generated.

In order to obtain a biased stratified sample, we simulated a two-phase random selection process from the population (Figure 1(a)) such that $P(Y = 1 \mid S) = 0.5$ and $P(X_e = 1 \mid S) = 0.5$. In a first step, an equal number of observations were randomly taken with $x_e = 1$ and with $x_e = 0$. In a second step, in each of these two strata from the first step, an equal number of observations with $y = 1$ and $y = 0$ were selected. By this, we partitioned the population into four equally sized strata corresponding to $(y, x_e) \in \{(1, 1), (1, 0), (0, 1), (0, 0)\}$.

Test data sets of size 10^4 were created in exactly the same way as the population. For our simulation study, we generated the population data set, the stratified data set, and the test set 1000 times for each feature distribution assumption. This way, we could empirically assess the variability of the performance of the correction and classification methods.

Application of Classifiers. We apply the seven correction approaches (Section 3.1) combined with the four considered classifiers (Section 3.2) to the synthetic data. To that end, stochastic IP oversampling and parametric IP bagging, proposed by us (Section 3.1.2), require a distribution assumption for the main features $\widetilde{\mathbf{X}}$. We always assume them to be normally distributed, even if the features in fact follow a Student's t-, Poisson, or Bernoulli distribution. We aim to find out how the algorithms get affected when assumptions are not met.

In fact, the four different distribution scenarios meet the Gaussian assumption in decreasing order: The normal distribution trivially fulfills it. The t-distribution is still continuous and symmetric so that the violation of the normality assumption may not get too severe. The Poisson distribution is discrete but approximately normal for $\lambda \geq 30$; however, in order to guarantee the normality assumption to be violated, we let $\lambda_i \in \{1, 2, 3, 4, 5\}$. The Bernoulli distribution cannot be seen as continuous and violates the normality assumption the most.

Evaluation. We measure the performance of the different classifiers combined with the various correction approaches by the Area-under-the-Receiver-Operating-Characteristic curve (AUC) [29]. The AUC is appropriate especially in the context of sample selection bias since it does not require binary prediction (i.e., discretizing continuous risks by choosing a cut-off) and is unaffected by linear transformations of the predictions as only ranks are considered. Thus, differences in performance should not be influenced by good or bad calibration of the prediction.

The goal of the comparison is to see whether correction approaches perform significantly better than not correcting. For each classifier, we fit a linear regression model with the AUC as target variable and the correction approach as covariate. The latter variable is dummy-coded with "no correction" as reference category. An approach is determined to differ significantly from the noncorrection approach if its coefficient's t-test confidence interval does not contain zero. For all comparisons, we use a level of significance of $\alpha = 5\%$.

Software. We used the statistical software R for all analyses [30]. More specifically, for building logistic regression models, we used the R package *stats* [30], for random forest the R package *ranger* [31], and for naive Bayes the R package *e1071* [32]. The modified implementation of the SMOTE algorithm is based on the R package *smotefamily* [33]. We validated our results via ROC analysis, using the R packages *pROC* [34] and *ROCR* [35].

4.2. Results. The simulation study yielded the following results (see also Figures 3–6): As expected, for every distribution scenario (see previous subsection) and all classifiers, the performance of learning on the entire population was significantly better than learning without correction on the smaller biased learning data set. Also, for all classifiers and in all distribution scenarios, there was at least one correction technique that outperformed the noncorrection approach (with two exceptions: logistic regression with additional interaction terms and naive Bayes, both in case of normally distributed main features).

However, there were differences between classifiers concerning the success of correction approaches. We start by contrasting logistic regression and the random forest as this comparison is of our primary interest.

The overall result for logistic regression (Figure 3) is that all correction approaches perform significantly better than noncorrection. Exceptions are costing and modified SMOTE in the normal distribution scenario which on average performs better than noncorrecting, but not significantly. For t-distributed and Poisson distributed features, the difference between the performance of noncorrection and the other approaches is more prominent than for the normal distribution scenario. In the Bernoulli case, this difference is the highest. Within each distribution scenario, the correction approaches perform similarly to each other.

For the random forest, the picture is rather different (Figure 4): Only one correction approach performs significantly better than noncorrecting: the parametric IP bagging proposed in this paper. In fact, for normally and t-distributed features, all other correction methods perform even worse than noncorrecting. In the Poisson scenario, they perform either worse than noncorrection or equally fine (IP bagging and costing). Only in the scenario in which the assumption of having continuous main features (required by the approaches

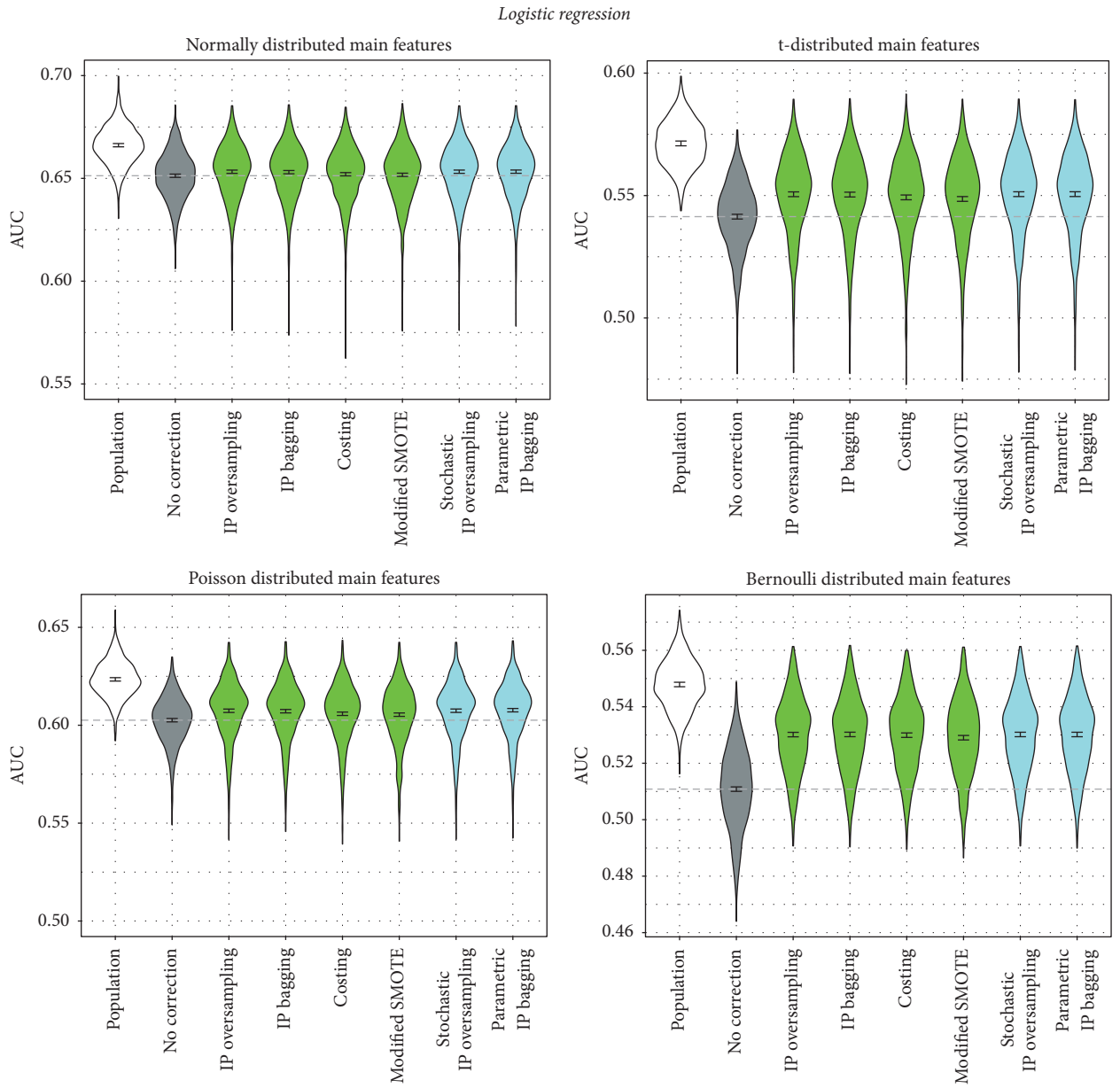


Figure 3: Performance of correction approaches in logistic regression, measured by AUC. We fit a linear model for the AUC as influenced by the correction method (dummy-coded, no correction as reference category). The graphic depicts 95% confidence intervals for the respective coefficients. The dashed line shows the intercept of the model (i.e., the mean AUC for no correction). The blue colored methods are newly proposed in this paper.

proposed by us) is not met at all (i.e., for the Bernoulli distribution) do almost all correction approaches perform better than not correcting. An exception is stochastic IP oversampling proposed by us. This approach failed in all distribution scenarios for the random forest.

Table 1 summarizes the properties of the correction approaches (Section 3.1.3) together with the just described results. We label the performance of an approach to be sufficient if it results in a significant increase of the AUC as compared to the noncorrection approach for the normal distribution scenario. Costing and modified SMOTE do not yield unambiguous improvements for logistic regression since their confidence intervals slightly overlap with the value under the null hypothesis. However, as we will see in Section 5, both approaches perform significantly better than noncorrection on real data.

In order to obtain a more comprehensive picture of the benefit of correcting for sample selection bias, we applied the correction methods in combination with two more classifiers, logistic regression with additional two-way interaction terms in addition to the linear terms and naive Bayes, leading to the following results.

Logistic regression with interaction terms yields a similar picture as standard logistic regression (Figure 5): All correction approaches perform similarly to each other. In the t- and Bernoulli scenario, again all correction approaches

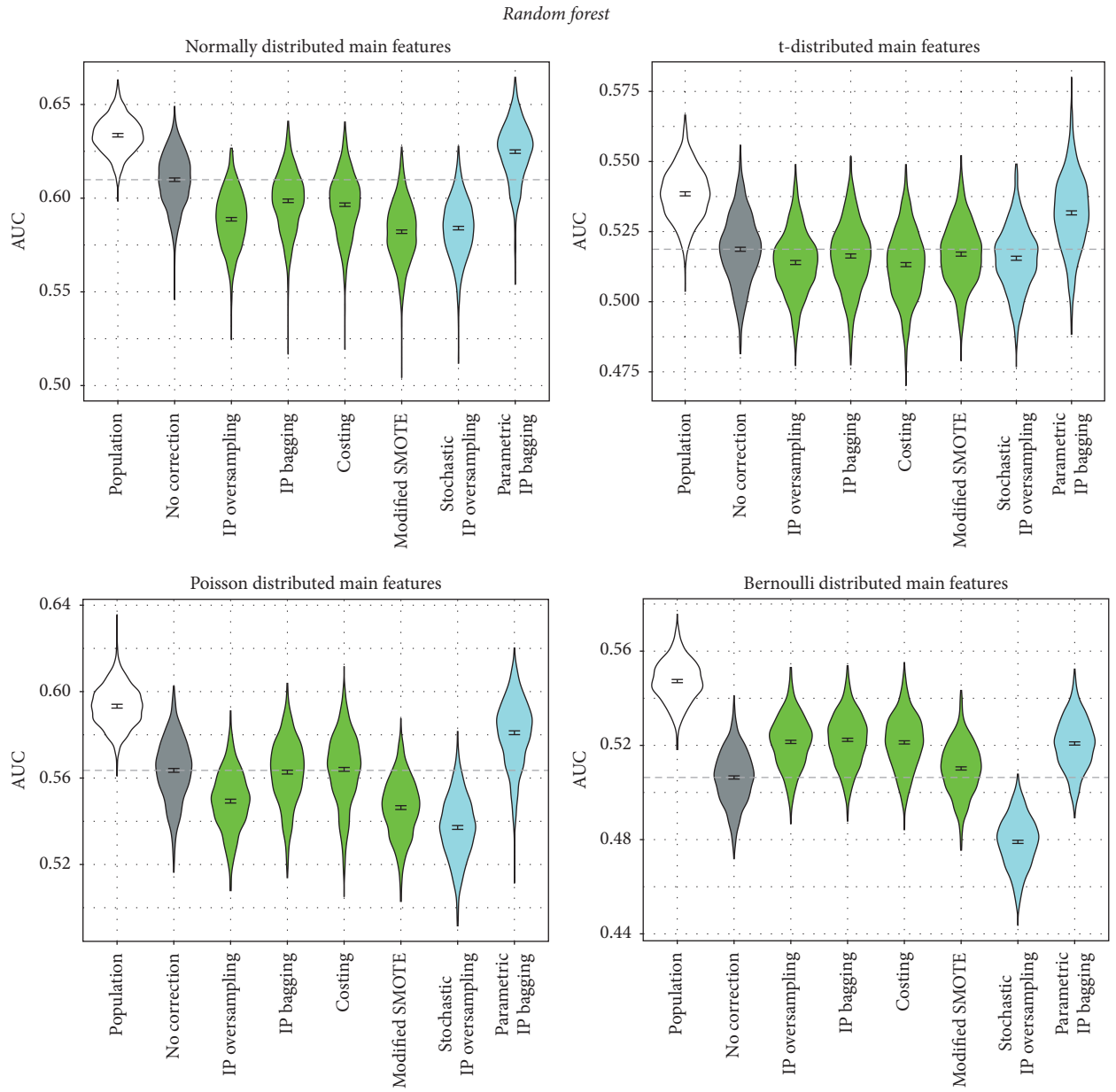


Figure 4: Performance of correction approaches in the random forest, measured by AUC. We fit a linear model for the AUC as influenced by the correction method (dummy-coded, no correction as reference category). The graphic depicts 95% confidence intervals for the respective coefficients. The dashed line shows the intercept of the model (i.e., the mean AUC for no correction). The blue colored methods are newly proposed in this paper.

outperform the noncorrection approach, except for costing for t-distributed features, which performs similarly to non-correcting. For both the normal and the Poisson distribution, all correction approaches perform significantly worse than not correcting. An exception is parametric IP bagging: Similar to the random forest case, only this method performs significantly better than no correction for the Poisson distribution scenario. For the normal distribution, the approach is the only one which does not perform significantly worse than the noncorrecting approach.

For naive Bayes (Figure 6), again all correction approaches behave similarly as in logistic regression. Depending on the data distribution, correction approaches perform worse or better than noncorrection. Especially in the normal distribution scenario, the correction approaches are not successful.

5. Real Data Application

This section investigates the performance of the correction methods in a real data example. Other than in the synthetic

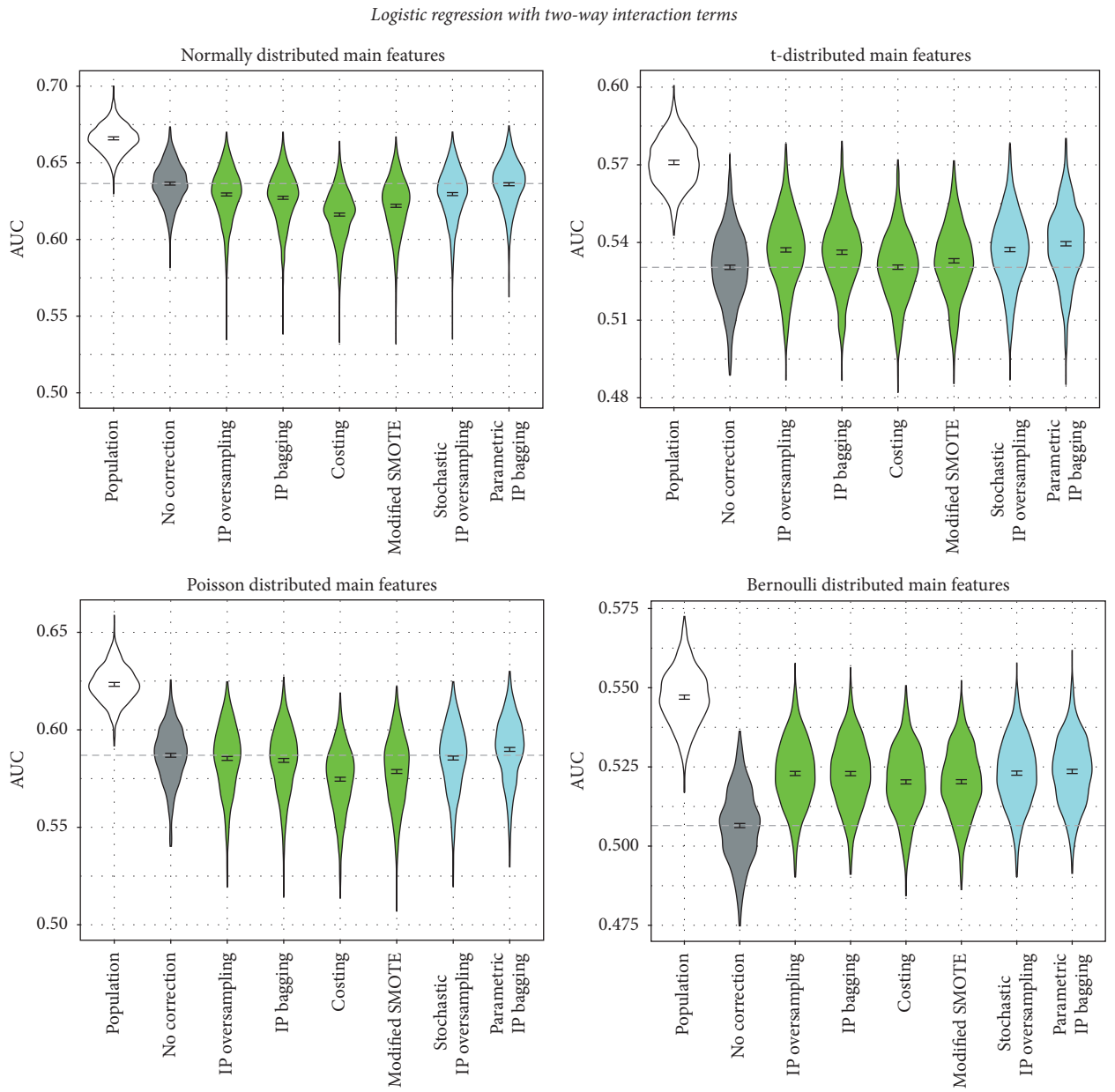


Figure 5: Performance of correction approaches in logistic regression with additional two-way interaction terms, measured by AUC. We fit a linear model for the AUC as influenced by the correction method (dummy-coded, no correction as reference category). The graphic depicts 95% confidence intervals for the respective coefficients. The dashed line shows the intercept of the model (i.e., the mean AUC for no correction). The blue colored methods are newly proposed in this paper.

data situation in the previous section, we do not know the true distribution of the entire population here. In order to still be able to evaluate the predictions appropriately, we chose a very large real data set from which we could extract a small stratified learning set and a large unbiased test set as described in the following.

5.1. Design

Data. We evaluate the various prediction methods on the example of the *hepatitis* data set (data ID: 269, exact name: "BNG (hepatitis)," version: 1) from OpenML [36]. It contains 10^6 observations of a binary outcome Y and 20 features. Y captures whether a hepatitis patient stayed alive and hence takes the categories *live* and *die*. We chose the binary variable *sex* as stratum feature X_e. From the remaining variables, we took into account the four continuous features *albumin*, *alkaline phosphatase*, *prothrombin time*, and *age*, denoted by $\widetilde{\mathbf{X}}$. These features were approximately normally distributed (partly after transformation; see the quantile-quantile plots in Figure 7) and strongly associated with the outcome.

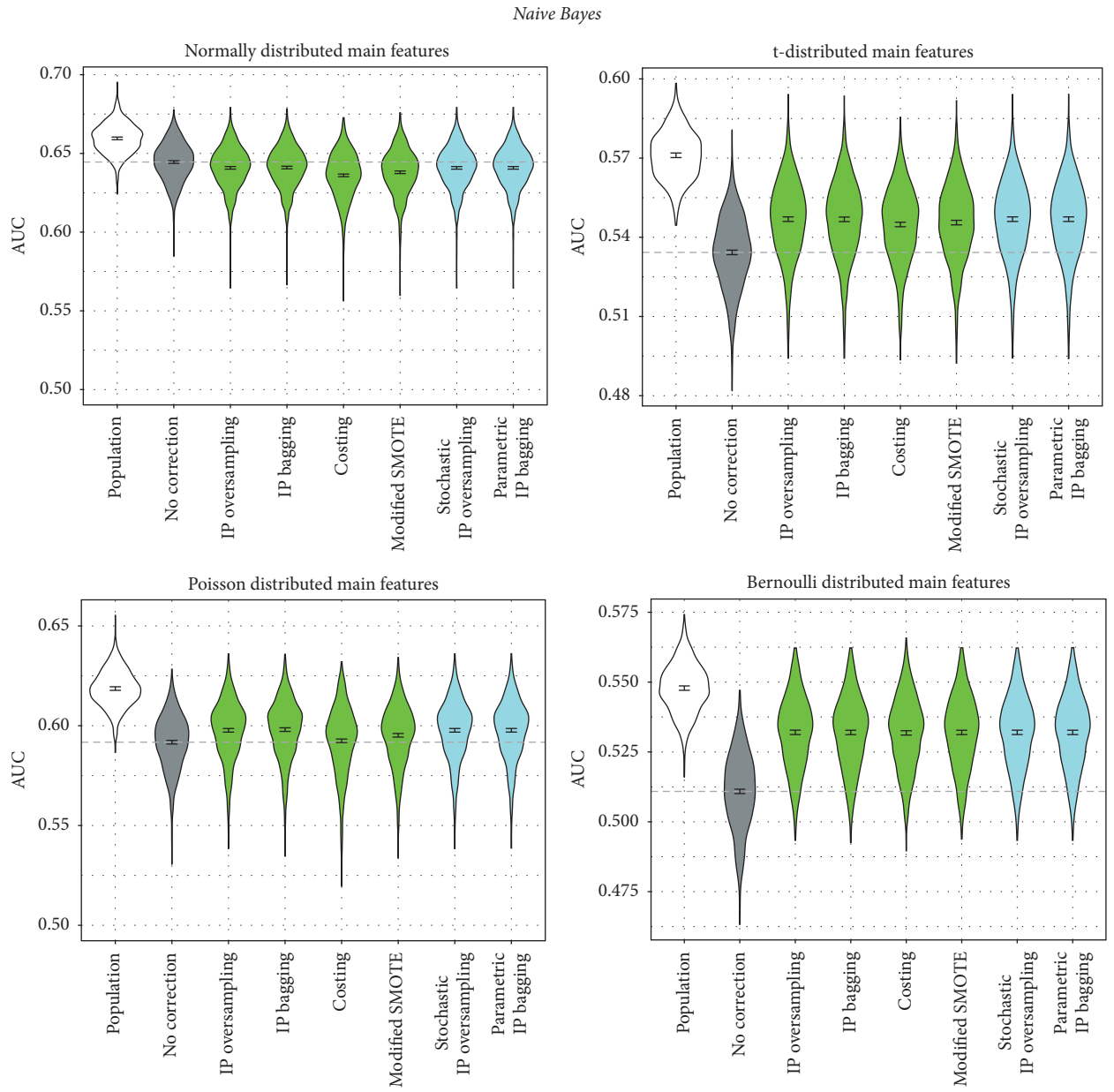


Figure 6: Performance of correction approaches in the naive Bayes classifier, measured by AUC. We fit a linear model for the AUC as influenced by the correction method (dummy-coded, no correction as reference category). The graphic depicts 95% confidence intervals for the respective coefficients. The dashed line shows the intercept of the model (i.e., the mean AUC for no correction). The blue colored methods are newly proposed in this paper.

Stratification Process. We aimed to evaluate the prediction methods on data sets which underwent sample selection bias. We hence constructed a learning data set by performing a two-phase stratified random selection process on the *hepatitis* data set. To that end, we selected $n = 2000$ out of the 10^6 observations, enriching the outcome Y and the feature variable *sex*, denoted by X_e. Figure 8 shows the sizes of the four strata in analogy to Figure 1(b). As test data set, we chose a subset of 10,000 observations from the hepatitis data set, disjoint to the learning data. We defined the first 10^6 observations (without the test data) as the population which served as reference learning data set as in the previous section.

5.2. Results. We trained all methods on the biased learning data and evaluated them on the unbiased test data. The resulting AUCs are compared by seven pairwise hypothesis tests according to [37]. We corrected for multiple testing via Bonferroni correction (i.e., set the threshold for p values to $\alpha^* = 0.05/7 = 0.0071$).

The real data results confirm the findings from the simulation study. For logistic regression, all weighting approaches

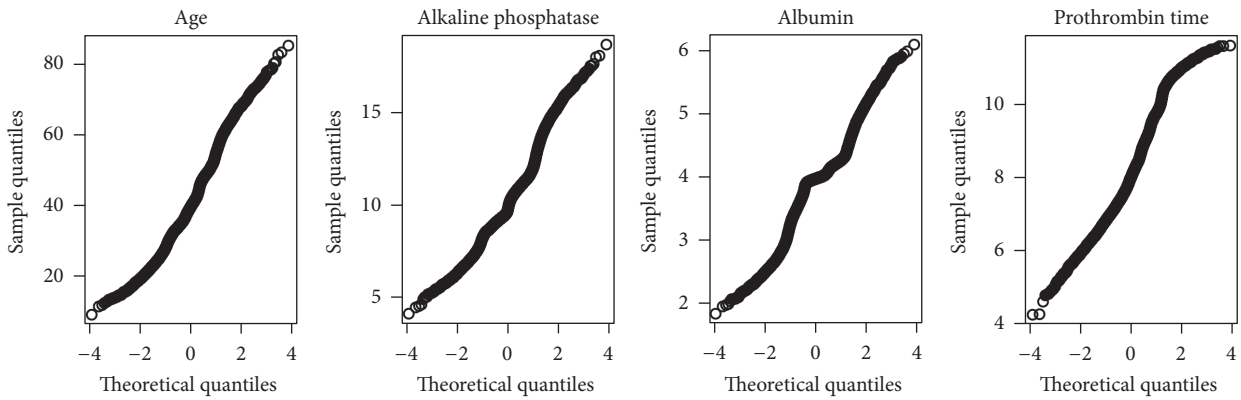


FIGURE 7: Normal quantile-quantile plots for main features $\widetilde{\mathbf{X}}$ in real data set. For visualization purposes, we only displayed a random sample of 10,000 observations instead of the full data set of size 10^6.

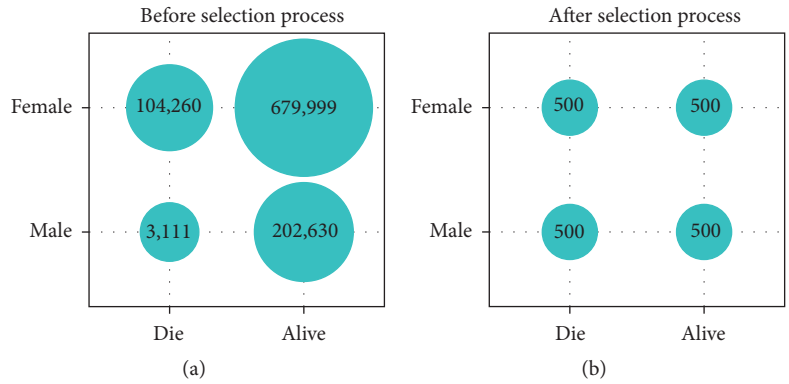


FIGURE 8: Cross table for the *hepatitis* data set before (a) and after (b) the selection process of a two-phase case-control study.

perform very similarly, which was significantly better than the nonweighting approach and even comparable to learning on a large population (Figure 9(a)).

For random forest, we obtain similar results as in the simulation study (Figure 9(b)): Only parametric IP bagging performs significantly better than the nonweighting approach. Costing and IP bagging perform insignificantly better; IP oversampling, modified SMOTE, and stochastic IP oversampling perform significantly worse.

Also, for logistic regression with interaction terms and naive Bayes, we obtain results matching with the simulation study: The assumptions for normality are met only roughly for the real data, in which case the correction approaches all perform similarly and better than no correction (Figure 9(d)).

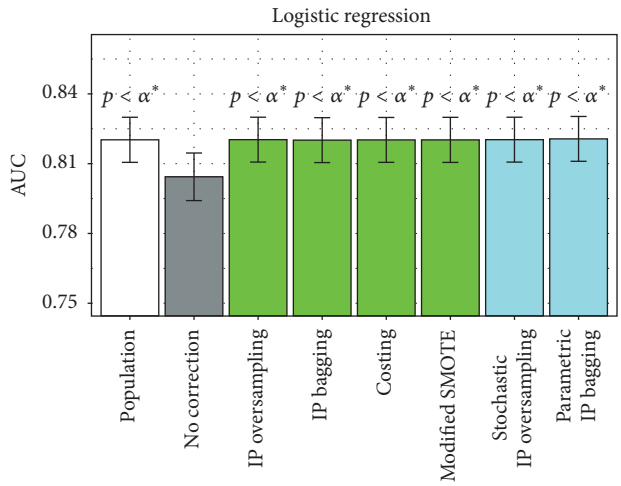
6. Discussion and Conclusion

We investigated how to learn classifiers on stratified random samples as resulting from two-phase case-control studies. Here, our emphasis was on random forest classification since previous bias correction methods did not pay special attention to resampling-based classifiers. However, we studied a broad range of classification techniques. This work hence guides the choice of such approaches also for other classifiers. The methods are immediately applicable due to the implementations provided in our R package *sambia*.

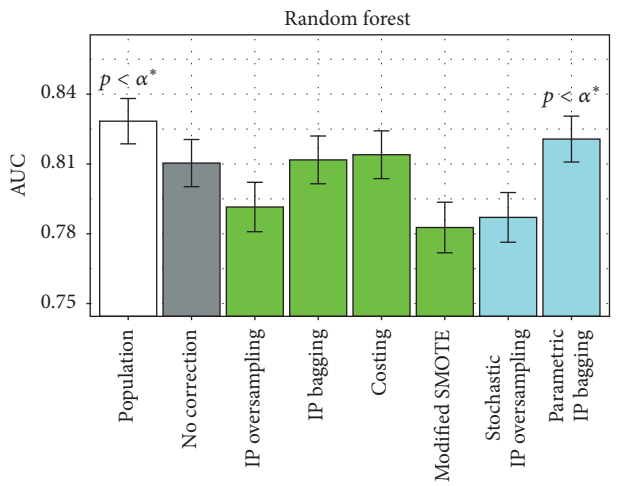
Both our simulation study and the real data application show that for classifiers trained on biased data sets prediction on unbiased data sets can be improved if the stratification process is taken into account and corrected for. However, state-of-the-art correction approaches from classical statistics (IP oversampling, IP bagging, costing, and modified SMOTE) do not yield the desired improvement for random forests. In fact, they can even lead to worse AUC values than those obtained when not performing any correction. From our two proposed approaches (stochastic IP oversampling and parametric IP bagging), on the other hand, the latter could always outperform the noncorrection approach.

We were also interested in all correction approaches' success when employed in the context of logistic regression. It turned out that any method improves prediction on an independent data set as compared to no correction, and all correction techniques perform similarly.

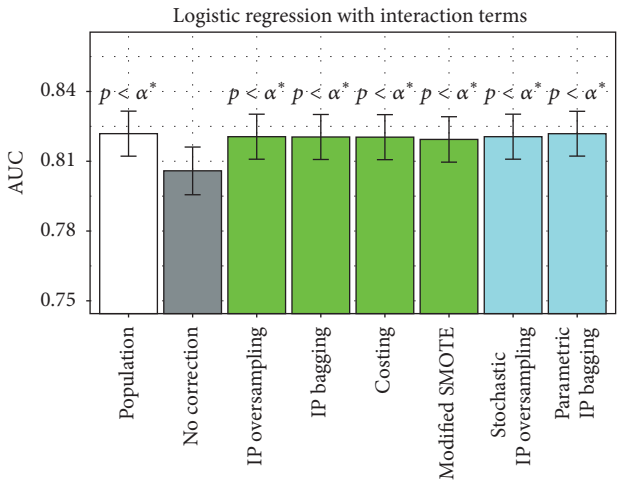
Table 1 helps to explain the different behaviors of the two classifiers: Correction approaches are based on one or several of the principles (i) IP weighting, (ii) rebuilding the original covariance structure, and (iii) increasing the number of learning observations as compared to the stratified sample. Obviously, weighting (Property (i)) should be applied in



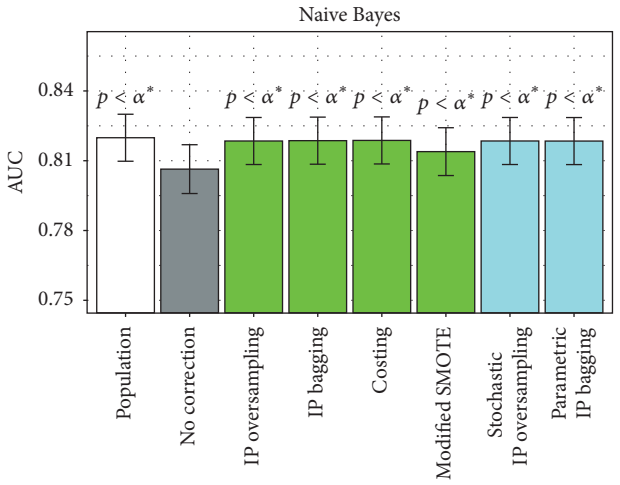
(a) Performance of logistic regression on real data. The graphic depicts 95% confidence intervals for the respective AUC value calculated and on the basis of [37]. All correction approaches perform similarly and significantly better than no correction (test by [37], $\alpha^* = 0.0071$)



(b) Performance of random forest on real data. The graphic depicts 95% confidence intervals for the respective AUC value calculated and on the basis of [37]. Only one correction approach, our novel parametric IP bagging, performs significantly better than no correction (test by [37], $\alpha^* = 0.0071$)



(c) Performance of logistic regression with all two-way interaction terms on real data. The graphic depicts 95% confidence intervals for the respective AUC value calculated and on the basis of [37]. All correction approaches perform significantly better than no correction (test by [37], $\alpha^* = 0.0071$)



(d) Performance of naive Bayes on real data. The graphic depicts 95% confidence intervals for the respective AUC value calculated and on the basis of [37]. All correction approaches perform significantly better than no correction (test by [37], $\alpha^* = 0.0071$)

Figure 9

order to obtain any improvement in performance. Moreover, the covariance structure should be corrected for (Property (ii)) when applying a random forest. IP oversampling and partly modified SMOTE failed to fulfill this criterion. For logistic regression, in contrast, the covariance structure does not matter since point estimates of regression coefficients are not affected when the variance in the data is underestimated. Last, sample sizes (Property (iii)) seem to matter more for random forests than for logistic regression. This is reasonable since too small sample sizes can restrict the range of the values of a feature and thus underestimate their variance leading to the same issue as for Property (ii). This made IP bagging and costing perform poorly for the random forest. This leaves us with stochastic IP oversampling and parametric IP bagging, both proposed in this paper. However, although stochastic IP oversampling was designed to fulfill Properties (i), (ii), and (iii), we could not yield successful results for random forests.

Having compared correction methods in random forests and in logistic regression, one may conclude that the choice of parametric IP bagging is advisable whenever the distribution assumptions for this approach are met. In order to once more revise this conclusion, we investigated the behaviors of all correction approaches in two more classifiers, a logistic regression model with additional interaction terms and the naive Bayes classifier. For the logistic regression model with interaction terms, once again only the parametric IP bagging

consistently outperformed the noncorrection approach. For naive Bayes, all approaches performed similarly among each other, confirming the above stated rule.

Against our expectations, naive Bayes failed in the simulation study for the normal distribution scenario but did well for all other distributions. A generally unexpected result was the poor accomplishment of stochastic IP oversampling. It performed worse than noncorrection in several scenarios and was successful only in those situations where all other correction approaches were successful as well.

For a random forest, parametric IP bagging is an effective technique for prediction on an unbiased data set and can also be preferred for other classifiers. However, in this paper, we restricted our simulations and real data example to the case where the main features could be assumed to be roughly normally distributed (after transformation, if necessary) so that the assumption of a multivariate normal distribution was appropriate. The success of parametric IP bagging generally depends on meeting the assumptions about the distributions of the features. Hence, the method should be chosen with care. On the other hand, our simulations show that, even in scenarios where assumptions are barely met (e.g., for Poisson distributed features), the approach still works. Clearly, one could also adjust the distribution family for the parametric bootstrap in parametric IP bagging. Even mixture distributions are conceivable (e.g., for bimodal feature distributions).

So far, parametric IP bagging has not been designed for binary or categorical main features or combinations of different types. This could be done by subgrouping the corresponding categories (or combining categories in the case of several categorical features) and estimating parameters in each of the subgroups for the assumed distribution family analogously to what we did for the different strata. Again, one would draw parametric bootstrap samples within all subgroups and construct a new unbiased sample within the scope of parametric IP bagging.

Even though our new approaches were developed for the random forest, they are generally tailored towards learning by any classifier and can be incorporated in other machine learning algorithms. Parametric IP bagging has been shown to perform well even if theoretical assumptions are not met. It can be applied on any stratified random sample and is not restricted to two-phase case-control studies. More generally, it is suited for any sample suffering from sample selection bias where the stratum features are categorical and the remaining features roughly follow a multivariate distribution from which parametric bootstrap samples can be drawn. For general classifiers, its performance is mostly comparable to that of other correction methods. Parametric IP bagging is the first correction method designed for the random forest and in that context clearly outperforms all other approaches.

Appendix

A. Dependence of S on X and Y for Label and Feature Bias

Label bias does not imply that S is independent of $\mathbf{X}$; that is,

$$P(S \mid \mathbf{X}, Y) = P(S \mid Y) \wedge P(S \mid Y) \neq P(S) \nRightarrow P(S \mid \mathbf{X}) = P(S). \tag{A.1}$$

Proof. Let $\mathbf{x} \coloneqq t(y)$, where t is a function mapping to $\{0, 1\}$. Then, $P(S \mid \mathbf{x}) = P(S \mid t(y)) = P(S \mid y) \neq P(S)$. □

Analogously, one can show that feature bias does not imply that S is independent of Y.

B. Covariance Matrix of Noise in Stochastic IP Oversampling

Here, we derive an appropriate noise covariance matrix to be added to the features $\widetilde{\mathbf{X}}_h'$ resulting from IP oversampling.

For one stratum h, we look at the covariance of the pair of features $\widetilde{X}_h^{(k)}$, $\widetilde{X}_h^{(j)}$ for $k, j \in \{1, \ldots, p\}$. For sample size n, we get per stratum a sample covariance per pair $\tilde{x}_h^{(k)}$, $\tilde{x}_h^{(j)}$, given by

$$s_{\tilde{x}_h^{(k)}, \tilde{x}_h^{(j)}} = \frac{1}{n_h - 1} \sum_{i=1}^{n_h} \left(\tilde{x}_{hi}^{(k)} - \overline{x}_h^{(k)}\right)\left(\tilde{x}_{hi}^{(j)} - \overline{x}_h^{(j)}\right), \tag{B.1}$$

where $\overline{x}_h^{(l)} \coloneqq (1/n_h) \sum_{i=1}^{n_h} \tilde{x}_{hi}^{(l)}$ for any $l \in \{1, \ldots, p\}$.

For IP oversampling, we replicate the data points by the factor w_h, which varies per stratum. Thus, the covariance of the modified sample is

$$\begin{aligned} s'_{\tilde{x}_h^{(k)}, \tilde{x}_h^{(j)}} &= \frac{1}{w_h n_h - 1} \sum_{i=1}^{n_h} w_h \left(\tilde{x}_{hi}^{(k)} - \overline{x}_h^{(k)}\right)\left(\tilde{x}_{hi}^{(j)} - \overline{x}_h^{(j)}\right) \\ &= \frac{w_h (n_h - 1)}{w_h n_h - 1} s_{\tilde{x}_h^{(k)}, \tilde{x}_h^{(j)}}. \end{aligned} \tag{B.2}$$

In addition to simple IP oversampling, stochastic IP oversampling incorporates the summation of some noise (matrix) $\tilde{\boldsymbol{\varepsilon}}$. We want the following to hold for a pair of the random vectors $\tilde{\boldsymbol{\varepsilon}}^{(k)}$, $\tilde{\boldsymbol{\varepsilon}}^{(j)}$ of size n_h:

$$\operatorname{cov}\left(\widetilde{X}_h^{(k)\prime} + \tilde{\boldsymbol{\varepsilon}}_h^{(k)}, \widetilde{X}_h^{(j)\prime} + \tilde{\boldsymbol{\varepsilon}}_h^{(j)}\right) = \operatorname{cov}\left(\widetilde{X}_h^{(k)}, \widetilde{X}_h^{(j)}\right), \tag{B.3}$$

where $\widetilde{X}_h^{(k)\prime}$, $\widetilde{X}_h^{(j)\prime}$ are the random variables resulting from replication by a factor w_h (oversampling).

We can simplify

$$\begin{aligned} &\operatorname{cov}\left(\widetilde{X}_h^{(k)\prime} + \tilde{\boldsymbol{\varepsilon}}_h^{(k)}, \widetilde{X}_h^{(j)\prime} + \tilde{\boldsymbol{\varepsilon}}_h^{(j)}\right) \\ &\quad = \operatorname{cov}\left(\widetilde{X}_h^{(k)\prime}, \widetilde{X}_h^{(j)\prime}\right) + \operatorname{cov}\left(\widetilde{X}_h^{(k)\prime}, \tilde{\boldsymbol{\varepsilon}}_h^{(j)}\right) \\ &\qquad + \operatorname{cov}\left(\widetilde{X}_h^{(j)\prime}, \tilde{\boldsymbol{\varepsilon}}_h^{(k)}\right) + \operatorname{cov}\left(\tilde{\boldsymbol{\varepsilon}}_h^{(k)}, \tilde{\boldsymbol{\varepsilon}}_h^{(j)}\right) \\ &\quad = \operatorname{cov}\left(\widetilde{X}_h^{(k)\prime}, \widetilde{X}_h^{(j)\prime}\right) + \operatorname{cov}\left(\tilde{\boldsymbol{\varepsilon}}_h^{(k)}, \tilde{\boldsymbol{\varepsilon}}_h^{(j)}\right), \end{aligned} \tag{B.4}$$

since the noise component $\tilde{\boldsymbol{\varepsilon}}_h^{(j)}$ should not correlate with the feature random vector $\mathbf{X}_k$ (neither $\tilde{\boldsymbol{\varepsilon}}_h^{(k)}$ with $\widetilde{X}_h^{(j)}$, resp.). This also holds for $j = k$.

We can estimate the components of the covariance matrix $\operatorname{cov}(\widetilde{X}_h^{(k)\prime}, \widetilde{X}_h^{(j)\prime})$ by $s'_{\tilde{x}_h^{(k)}, \tilde{x}_h^{(j)}} = (w_h(n_h - 1)/(w_h n_h - 1)) s_{\tilde{x}_h^{(k)}, \tilde{x}_h^{(j)}}$.

Substituting this into (B.3) yields, for the entries of our *noise covariance matrix*,

$$s'_{\tilde{\boldsymbol{\varepsilon}}_h^{(k)},\tilde{\boldsymbol{\varepsilon}}_h^{(j)}} = s_{\tilde{x}_h^{(k)},\tilde{x}_h^{(j)}} - \frac{w_h\left(n_h-1\right)}{w_h n_h - 1} s_{\tilde{x}_h^{(k)},\tilde{x}_h^{(j)}} = \frac{w_h - 1}{w_h n_h - 1} s_{\tilde{x}_h^{(k)},\tilde{x}_h^{(j)}}. \tag{B.5}$$

In terms of random variables, the empirical covariance matrix combining all entries $s'_{\tilde{\boldsymbol{\varepsilon}}_h^{(k)},\tilde{\boldsymbol{\varepsilon}}_h^{(j)}}$ for all $k, j \in \{1,\ldots,p\}$ would be replaced by Σ_h^{adj} and the empirical covariance matrix combining all entries $s_{\tilde{x}_h^{(k)},\tilde{x}_h^{(j)}}$ for all $k, j \in \{1,\ldots,p\}$ by Σ_h.

Additional Points

Supplementary Material. Additional figures, code, and data are available at https://www.helmholtz-muenchen.de/index.php?id=47085.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Christiane Fuchs and Fabian J. Theis are supported by the German Research Foundation (DFG) within the Collaborative Research Centre 1243, Subproject A17.

References

[1] C. E. Rossiter and J. J. Schlesselman, "Case-Control Studies. Design, Conduct, Analysis.," *Biometrics*, vol. 39, no. 3, p. 821, 1983.

[2] E. W. Steyerberg, G. J. J. M. Borsboom, H. C. van Houwelingen, M. J. C. Eijkemans, and J. D. F. Habbema, "Validation and updating of predictive logistic regression models: A study on sample size and shrinkage," *Statistics in Medicine*, vol. 23, no. 16, pp. 2567–2586, 2004.

[3] Y. Huang and M. S. Pepe, "Assessing risk prediction models in case-control studies using semiparametric and nonparametric methods," *Statistics in Medicine*, vol. 29, no. 13, pp. 1391–1410, 2010.

[4] S. Rose and M. van der Laan, "A Note on Risk Prediction for Case-Control Studies, 2008," in press.

[5] K. J. M. Janssen, Y. Vergouwe, C. J. Kalkman, D. E. Grobbee, and K. G. M. Moons, "A simple method to adjust clinical prediction models to local circumstances," *Canadian Journal of Anesthesia*, vol. 56, no. 3, pp. 194–201, 2009.

[6] J. E. White, "A two stage design for the study of the relationship between a rare exposure and a rare disease," *American Journal of Epidemiology*, vol. 115, no. 1, pp. 119–128, 1982.

[7] J. M. Satagopan, E. S. Venkatraman, and C. B. Begg, "Two-stage designs for gene-desease association studies with sample size constraints," *Biometrics. Journal of the International Biometric Society*, vol. 60, no. 3, pp. 589–597, 2004.

[8] O. Saarela, S. Kulathinal, and J. Karvanen, "Secondary analysis under cohort sampling designs using conditional likelihood," *Journal of Probability and Statistics*, Article ID 931416, 2012.

[9] T. Saidel, R. Adhikary, M. Mainkar et al., "Baseline integrated behavioural and biological assessment among most at-risk populations in six high-prevalence states of India: Design and implementation challenges," *AIDS*, vol. 22, no. 5, pp. S17–S34, 2008.

[10] T. C. Mills, R. Stall, L. Pollack et al., "Health-related characteristics of men who have sex with men: A comparison of those living in "gay ghettos" with those living elsewhere," *American Journal of Public Health*, vol. 91, no. 6, pp. 980–983, 2001.

[11] C. Kendall, L. R. F. S. Kerr, R. C. Gondim et al., "An empirical comparison of respondent-driven sampling, time location sampling, and snowball sampling for behavioral surveillance in men who have sex with men, Fortaleza, Brazil," *AIDS and Behavior*, vol. 12, no. 1, pp. S97–S104, 2008.

[12] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proceedings of the 21th International Conference on Machine Learning (ICML '04)*, pp. 903–910, Alberta, Canada, July 2004.

[13] J. J. Heckman, "Sample selection bias as a specification error," *Econometrica*, vol. 47, no. 1, pp. 153–161, 1979.

[14] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh, "Sample selection bias correction theory," in *Algorithmic learning theory*, vol. 5254 of *Lecture Notes in Comput. Sci.*, pp. 38–53, Springer, Berlin, 2008.

[15] G. King and L. Zeng, "Logistic regression in rare events data," *Political Analysis*, vol. 9, no. 2, pp. 137–163, 2001.

[16] T. Lumley, "Analysis of complex survey samples," *Journal of Statistical Software*, vol. 9, pp. 1–19, 2004.

[17] W. H. Dumouchel and G. J. Duncan, "Using sample survey weights in multiple regression analyses of stratified samples," *Journal of the American Statistical Association*, vol. 78, no. 383, pp. 535–543, 1983.

[18] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM '03)*, pp. 435–442, Melbourne, Fla, USA, November 2003.

[19] W. Fan and I. Davidson, "On sample selection bias and its efficient correction via model averaging and unlabeled examples," in *Proceedings of the 7th SIAM International Conference on Data Mining (SIAM '07)*, pp. 320–331, Minneapolis, Minn, USA, April 2007.

[20] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI '01)*, pp. 973–978, New York, NY, USA, August 2001.

[21] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe," *Journal of the American Statistical Association*, vol. 47, pp. 663–685, 1952.

[22] J. M. Robins, A. Rotnitzky, and L. P. Zhao, "Estimation of regression coefficients when some regressors are not always observed," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 846–866, 1994.

[23] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[24] M. Nahorniak, D. P. Larsen, C. Volk, and C. E. Jordan, "Using inverse probability bootstrap sampling to eliminate sample induced bias in model based analysis of unequal probability samples," *PLoS ONE*, vol. 10, no. 6, Article ID e0131765, 2015.

[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[26] L. Fahrmeir, T. Kneib, and S. Lang, "Regression," in *Statistik und ihre Anwendungen*, Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, 2009.

[27] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, New York, NY, USA, 2001.

[29] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[30] R. Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015.

[31] M. N. Wright and A. Ziegler, "ranger: a fast implementation of random forests for high dimensional data in C++ and R," *Journal of Statistical Software*, vol. 77, no. 1, pp. 1–17, 2017.

[32] D. Meyer, E. K. Dimitriadou, A. Hornik, Weingessel., and F. Leisch, *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien, Vienna, Austria, 2015.

[33] W. Siriseriwan, "smotefamily: A Collection of Oversampling Techniques for Class Imbalance Problem Based on SMOTE," 2016.

[34] X. Robin, N. Turck, A. Hainard et al., "pROC: an open-source package for R and S+ to analyze and compare ROC curves," *BMC Bioinformatics*, vol. 12, no. 1, Article ID 77, 2011.

[35] T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer, "ROCR: visualizing classifier performance in R," *Bioinformatics*, vol. 21, no. 20, pp. 3940-3941, 2005.

[36] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "OpenML: Networked Science in Machine Learning," *SIGKDD Explorations*, vol. 15, no. 2, pp. 49–60, 2014.

[37] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach," *Biometrics*, vol. 44, no. 3, pp. 837–845, 1988.