Catalogue no. 12-001-X
ISSN 1492-0921

Survey Methodology

A method for estimating the effect of classification errors on statistics for two domains

by Yanzhe Li, Sander Scholtus and Arnout van Delden

Release date: January 3, 2024

Statistics Canada Statistique Canada

Canada

How to obtain more information

For information about this product or the wide range of services and data available from Statistics Canada, visit our website, www.statcan.gc.ca.

You can also contact us by

Email at infostats@statcan.gc.ca

Telephone, from Monday to Friday, 8:30 a.m. to 4:30 p.m., at the following numbers:

- Statistical Information Service 1-800-263-1136
- National telecommunications device for the hearing impaired 1-800-363-7629
- Fax line 1-514-283-9350

Standards of service to the public

Statistics Canada is committed to serving its clients in a prompt, reliable and courteous manner. To this end, Statistics Canada has developed standards of service that its employees observe. To obtain a copy of these service standards, please contact Statistics Canada toll-free at 1-800-263-1136. The service standards are also published on www.statcan.gc.ca under "Contact us" > "Standards of service to the public."

Note of appreciation

Canada owes the success of its statistical system to a long-standing partnership between Statistics Canada, the citizens of Canada, its businesses, governments and other institutions. Accurate and timely statistical information could not be produced without their continued co-operation and goodwill.

Published by authority of the Minister responsible for Statistics Canada

An HTML version is also available.

Cette publication est aussi disponible en français.

Survey Methodology, December 2023
Vol. 49, No. 2, pp. 569-605
Statistics Canada, Catalogue No. 12-001-X

A method for estimating the effect of classification errors on statistics for two domains

Yanzhe Li, Sander Scholtus and Arnout van Delden[1]

Abstract

Being able to quantify the accuracy (bias, variance) of published output is crucial in official statistics. Output in official statistics is nearly always divided into subpopulations according to some classification variable, such as mean income by categories of educational level. Such output is also referred to as domain statistics. In the current paper, we limit ourselves to binary classification variables. In practice, misclassifications occur and these contribute to the bias and variance of domain statistics. Existing analytical and numerical methods to estimate this effect have two disadvantages. The first disadvantage is that they require that the misclassification probabilities are known beforehand and the second is that the bias and variance estimates are biased themselves. In the current paper we present a new method, a Gaussian mixture model estimated by an Expectation-Maximisation (EM) algorithm combined with a bootstrap, referred to as the EM bootstrap method. This new method does not require that the misclassification probabilities are known beforehand, although it is more efficient when a small audit sample is used that yields a starting value for the misclassification probabilities in the EM algorithm. We compared the performance of the new method with currently available numerical methods: the bootstrap method and the SIMEX method. Previous research has shown that for non-linear parameters the bootstrap outperforms the analytical expressions. For nearly all conditions tested, the bias and variance estimates that are obtained by the EM bootstrap method are closer to their true values than those obtained by the bootstrap and SIMEX methods. We end this paper by discussing the results and possible future extensions of the method.

Key Words: Bias; Variance; Misclassification; Binary classifier; Gaussian mixture model; EM algorithm.

1. Introduction

Accurate published output is crucial, especially in official statistics (Eurostat, 2009, page 32), where most outcomes are used for policy making. One of the important errors affecting the accuracy of output are measurement errors in the variable that is used to group output into subpopulations. The simplest type of estimator for which the effect of misclassifications has been studied are proportions in contingency tables. Bross (1954) gave an expression for the bias of an estimated proportion of a binary variable in the case of misclassifications, which is a special case of expression (C.1) in Appendix C. A more complicated situation concerns the accuracy of level estimators (totals, means) and ratios thereof as affected by misclassifications. For instance, one may be interested in the average risk of poverty for working persons by categories of educational level (Eurostat, 2022), while part of those education values are misclassified. Estimating population parameters of a continuous variable in each class is referred to as domain statistics. The current study focuses on estimation of the bias and variance of totals or means as affected by misclassifications. There is a large amount of literature on misclassifications, for instance Buonaccorsi (2010), Keogh, Shaw, Gustafson, Carroll, Deffner, Dodd, Küchenhoff, Tooze, Wallace, Kipnis and Freedman (2020) and Shaw, Gustafson, Carroll, Deffner, Dodd, Keogh, Kipnis, Tooze, Wallace, Küchenhoff and Freedman (2020). Van den Hout and Van der Heijden (2002) provide an extensive overview of literature on bias and variance

1. Yanzhe Li, Statistics Netherlands, PO Box 24500, 2490 HA The Hague, The Netherlands; Sander Scholtus, Statistics Netherlands, PO Box 24500, 2490 HA The Hague, The Netherlands. E-mail: s.scholtus@cbs.nl; Arnout van Delden, Statistics Netherlands, PO Box 24500, 2490 HA The Hague, The Netherlands.

in the case of misclassifications. These references underline that quantifying the effect of misclassifications on population estimates is relevant for a wide range of disciplines, including medicine, epidemiology, statistical astronomy, sociology, land cover mapping, randomised response studies and data confidentiality. The latter two cases are special in the sense that their misclassifications are deliberately added to the data by a known mechanism.

Official statistics often make use of statistical registers to identify their target population and divide it into subpopulations. One example is a statistical business register containing a list of statistical units over time, with background variables such as economic activity and size class to stratify them into subpopulations (United Nations, 2015). Another example is a population register, with background variables such as date of birth, gender, place of residence and highest attained level of education (Bakker, Van Rooijen and Van Toor, 2014). These registers are often created with one or more administrative sources. Errors in the classification variables can occur due to errors during registration into those administrative sources, administrative delays, errors in the linkage of the administrative sources or changes in circumstances of the units which are not reported. Another source of error are differences between concepts used by register owners and those used in official statistics (Magnusson, Palm, Branden and Mörner, 2017). Sometimes classification variables are derived by applying machine learning algorithms [see, e.g., Meertens, Diks, Van den Herik and Takes (2020)] and errors made by those algorithms subsequently lead to misclassifications.

In practical situations where output is produced for official statistics, one tries to reduce the number of misclassifications for instance by automatic or manual editing. In the case of business statistics, economic activity codes are known to be prone to errors. In that case, manual editing is applied which is usually limited to the most influential units – the largest enterprises – while the remaining units are left uncorrected. One often hopes that errors in the smaller units do not have a large impact on the published figures. Unfortunately, this need not be true. Van Delden, Scholtus and Burger (2016) showed that in a particular case study, observed levels of misclassification in smaller and medium-sized enterprises resulted in considerable bias of turnover totals for some publication cells. The reason for this bias was that the inflow of turnover from units unjustly classified into the target class was not balanced by the outflow of turnover from units unjustly not classified into the target class. In the context of contingency tables, Schwartz (1985) underlined the importance of misclassifications on output accuracy by framing it as "a neglected problem".

Analytical expressions for the bias and variance of means or totals of subpopulations as affected by misclassifications have been published by Selén (1986) and Van Delden et al. (2016). Furthermore, the bias and variance of totals of subpopulations were discussed by Kooiman, Willenborg and Gouweleeuw (1997) in the context of data confidentiality. These analytical expressions have two disadvantages. The first disadvantage is that they rely on the assumption that the probabilities of all types of misclassification are known, or that they have been accurately estimated by means of a sample. In the context of data confidentiality these probabilities are known, because misclassifications are applied on purpose to avoid disclosure (Kooiman et al., 1997). In most applications, however, the probabilities of misclassification are

unknown. Estimated classification error probabilities are often obtained from previous knowledge or a comparable dataset (Edwards, Bakoyannis, Yiannoutsos, Mburu and Cole, 2019; Edwards, Cole and Fox, 2020), or from an audit sample (Gravel and Platt, 2018). Accurately estimating these probabilities using sufficiently large sample sizes requires a considerable amount of manual labour. The second disadvantage is that the bias and variance estimates based on those analytical expressions are biased themselves. This is mentioned by Kooiman et al. (1997) and worked out in more detail by Van Delden et al. (2016). The main reason is that the expressions for the bias and variance depend on true population totals which are unknown. When estimating the bias and variance, these true population totals are replaced by their biased estimators of the population totals, leading to biased estimates of the bias and (to a lesser extent) of the variance of the totals. In the special case that one is only interested to estimate a proportion without a numerical variable, different methods to correct for bias can be found in Kloos, Meertens, Scholtus and Karch (2021).

As an alternative to the use of analytical expressions, numerical approaches have been developed. The bootstrap approach may be used to estimate output accuracy in the case of misclassifications. Zhang (2011) used the bootstrap method to measure the variance of observed totals of subpopulations caused by misclassification of households. Van Delden et al. (2016) presented a bootstrap approach to estimate the bias and variance of means or totals of subpopulations. The bootstrap approach is very flexible and can be adapted to many estimators. Unfortunately, the bootstrap approach has the same two disadvantages as the analytical approach: it requires that the probabilities of misclassification are accurately known and, in general, it leads to biased estimates of bias and variance. In fact, for simple domain parameters such as totals and proportions, the bootstrap method and the analytical expressions lead to near-identical results; see for instance Van Delden et al. (2016). For non-linear domain parameters such as ratios, the bootstrap results are generally more accurate in the case of skewed distributions because some underlying assumptions of the approximation used in the analytical bias and variance expressions might be violated; see Van Delden, Scholtus, Burger and Meertens (2023).

Another numerical method that has been proposed for misclassifications is the SIMEX ("SIMulation and Extrapolation") method, which was first introduced by Cook and Stefanski (1994). It was developed to estimate the impact of measurement errors and is adapted in the field of misclassification by Küchenhoff, Mwalili and Lesaffre (2006) and Hopkins and King (2010). Similarly to the bootstrap approach, the SIMEX method starts with applying misclassifications to the observed data after which the estimators (totals, means) are recalculated. Additionally, the SIMEX method applies a simulation process that introduces multiple sets of extra errors to estimate the effect of misclassifications. Finally, one extrapolates the estimates to the error-free condition. Here we have applied the ideas behind this SIMEX procedure to test whether it can be used to overcome the bias in the estimates of the bias and variance of means and totals by misclassifications.

In the current study, we present a new method that uses a mixture model to estimate the bias and variance of means and totals by misclassifications. An Expectation-Maximisation (EM) algorithm is used to estimate the mixture model. Since the true classes of units are unknown in the case of misclassifications, these true classes can be regarded as an unobserved ("latent") variable in the mixture model. We model the numerical

target variable in each class as a mixture of different normal distributions (a Gaussian mixture), which can also accommodate target variables that do not have a normal distribution; see McLachlan and Peel (2000). The method is referred to as "the EM bootstrap method". In the current study we limit ourselves to a binary classification variable. Our approach can be extended to a situation with multiple classes, which is further treated in the discussion.

The new method performs better with respect to the two mentioned disadvantages than the bootstrap and the SIMEX method. First of all, it does not require an estimate of the misclassification probabilities beforehand, although in more complicated cases it is computationally efficient to have a rough estimate of the misclassification rates since it can be used as an informed start for the EM algorithm. Second, we will show that – at least in the binary case – the new method leads to more accurate estimates of the bias and variance of means and totals affected by misclassifications than the bootstrap and the SIMEX method.

In the current paper we evaluate the accuracy of bias and variance estimates under misclassification, by comparing the proposed EM bootstrap method with the bootstrap method and with the adapted SIMEX method (referred to as "the SIMEX bootstrap method") in a simulation study and in a case study. In both studies, the proportions of two classes are varied as well as the misclassification rates. In the simulation study the data distribution in each class is a Gaussian mixture. The case study uses empirical distributions from a dataset with log turnover per enterprise in the Netherlands. For the empirical data we use a Gaussian mixture model in which the optimal number of components has to be estimated. In both studies we compare true bias and variance values with their estimates.

The remainder of this paper is organised as follows. Section 2 describes the details of the three methods. The evaluation of the methods for the simulation study is given in Section 3 and for the case study in Section 4. Section 5 discusses the results of our study and proposes future directions. Furthermore, Appendices A, B and C provide additional material. Appendix A discusses three different extrapolation functions in the SIMEX method. Appendix B compares the difference of using BIC and sBIC as a criterion for selecting the optimal number of components in the case study. Appendix C includes some theoretical properties of the bootstrap and the EM bootstrap method. R code that implements the methods used in this study can be found at https://github.com/Yanzhee/EM-bootstrapping.

2. Methodology

2.1 General settings

We consider a situation where a population of N units is classified into two classes. The indicator variable for the true membership of one of these classes is denoted as $z \in \{0, 1\}$. For convenience, from now on we will refer to the classes of interest as class 1 and class 0 and use the indicator z interchangeably with the classification itself. The values of z are assumed to contain no classification errors and are considered to be fixed for the population of interest. In practice, we do not know the true classes of all units. We can

only infer them through, for example, manual checking by experts, automatic classification from multiple machine-learning algorithms, etc. The true proportion of class 1 in the population is denoted by α_1; the true proportion of class 0 is $1-\alpha_1$.

The observed classification variable is denoted as $\hat{z} \in \{0,1\}$. In practice, the observed classes of units contain classification errors. The classification errors can be from misunderstandings of class definition, miscommunication or simply typos.

In practice, one is often interested in estimating population parameters of a continuous variable y in each class, referred to as domain statistics. For example, when y represents turnover of enterprises in various industries, the total turnover of each industry will be an interesting indicator. We use ζ to denote a true population parameter, based on y and the true classification variable z. In what follows, the continuous variable y is assumed to be error-free.

Examples of common domain parameters ζ that are included in our study are: the total sum of y for class 1 (T_1), the proportion of class 1 (α_1), and the standard deviation of y for class 1 (σ_1). Table 2.1 lists the formulas for ζ given variables y and z. Since our study is under the setting of a binary classifier, the accuracy of domain statistics in class 0 is directly related to the accuracy of the corresponding estimates in class 1.

Table 2.1
List of formulas for examples of domain parameters ζ.

Statistics of Interest	Notation	Formula given y and z
Total sum of y for class 1	T_1	$\sum_i z_i y_i$
Proportion of class 1	α_1	$\sum_i z_i / N$
Standard deviation of y for class 1	σ_1	$\sqrt{\frac{1}{\sum_i z_i}\sum_i z_i (y_i - \mu_1)^2}$

Notes: 1. i stands for a unit in the population.
2. $\mu_1 = \sum_i z_i y_i / \sum_i z_i = T_1/(N\alpha_1)$ is the mean of y for class 1.
3. By replacing z_i with $\hat{z}_i$ in the formulas, $\hat{T}_1$, $\hat{\alpha}_1$, and $\hat{\sigma}_1$ can be calculated.

The parameter ζ requires the true values of z and therefore cannot be computed in practice. An obvious estimator is obtained by replacing the unknown true z with the observed $\hat{z}$ in the expressions for ζ; this yields the domain statistic $\hat{\zeta}$. For example, by replacing z_i with $\hat{z}_i$ in Table 2.1, formulas are obtained for the domain statistics $\hat{T}_1$, $\hat{\alpha}_1$, and $\hat{\sigma}_1$. Note that the hat in $\hat{\zeta}$ signifies an estimator, whereas in $\hat{z}_i$ it has no meaning other than to distinguish the observed indicator from the true indicator z_i.

Our study uses bias and variance to measure the effects that classification errors bring to $\hat{\zeta}$. Bias is defined as the difference between the expected values of the estimated output and the true value of domain parameters. Variance is a measure of the expected amount that the estimated domain statistics will change if different classification variables with the same error distribution are used. Mathematically, we define:

$$\begin{aligned} \textbf{Bias} &= \mathrm{E}(\hat{\zeta} - \zeta), \\ \textbf{Variance} &= \mathrm{Var}(\hat{\zeta}) = \mathrm{E}\left(\left(\hat{\zeta} - \mathrm{E}(\hat{\zeta})\right)^2\right). \end{aligned} \tag{2.1}$$

Besides classification errors, for simplicity we assume that no other errors occur. In line with a common type of application in official statistics, we are interested here in bias and variance due to classification errors for a fixed finite population. That is to say, in (2.1) we implicitly condition on the realised values of $z_1, \ldots, z_N$ and $y_1, \ldots, y_N$.

2.2 Mixture model

2.2.1 Model setup

Our proposed new method to estimate the bias and variance of $\hat{\zeta}$ requires a model for the observed values $y_1, \ldots, y_N$ and $\hat{z}_1, \ldots, \hat{z}_N$. Here, we propose to use a mixture model, where the distributions in class $z = 1$ and class $z = 0$ may be different. For simplicity, we assume that the values of units $i = 1, \ldots, N$ are drawn independently of each other. In addition, we assume that classification errors in $\hat{z}_i$ occur with the same probabilities for all units, which also means that they are independent of the continuous variable y_i. The latter assumption allows us to model the observed values y_i and $\hat{z}_i$ separately, since it implies that, within each true class, the joint density of y_i and $\hat{z}_i$ is factorised as follows:

$$f(y_i, \hat{z}_i = b \mid z_i = a) = f(y_i \mid z_i = a) \cdot P(\hat{z}_i = b \mid z_i = a) \tag{2.2}$$

for all $i = 1, \ldots, N,$ with $a, b \in \{0, 1\}$. Here, $P(\hat{z} = b \mid z = a)$ denotes a classification error probability and $f(y \mid z = a)$ denotes the density of the continuous variable in class a. We will now describe these two parts of the model in more detail.

Probability matrix. The probabilities of classification errors are modelled by a 2×2 transition matrix $\mathbf{P}$ (Formula 2.3). It describes the relationship between the true classes z (rows) and the observed classes $\hat{z}$ (columns).

$$\mathbf{P} = \begin{pmatrix} p_{11} & 1 - p_{11} \\ 1 - p_{00} & p_{00} \end{pmatrix}. \tag{2.3}$$

The value p_{ab} indicates the probability of observing a unit in class b when its true class is a. Thus, for unit i, if its true class is 1, its probability to be observed in class 1 is $P(\hat{z}_i = 1 \mid z_i = 1) = p_{11},$ and its probability to be observed in class 0 is $P(\hat{z}_i = 0 \mid z_i = 1) = 1 - p_{11};$ similarly, for a unit in true class 0, $P(\hat{z}_i = 0 \mid z_i = 0) = p_{00}$ and $P(\hat{z}_i = 1 \mid z_i = 0) = 1 - p_{00}$. For reasonable classifiers, the values of p_{11} and p_{00} should be above 0.5.

Gaussian mixture model. We assume that the distribution of y for each class z follows a Gaussian mixture model. The number of Gaussian components in class 1 is $q_1,$ and the number in class 0 is q_0. Note that this means that the overall model for y can be seen as a "mixture of mixtures", with the first mixing

level given by the true class z and the second mixing level by the Gaussian mixture components within a true class.

For better explanation, a component variable m is defined to identify which component of the Gaussian mixture model each unit belongs to. The variable m is unobserved. The distribution of y_i depends on the class it belongs to (the value of z_i) and also which component in this class it belongs to (the value of m_i); we make the conventional assumption that each unit i belongs to a unique class-component pair (z_i, m_i). By the law of total probability, the density of y_i conditional on z_i is:

$$f\left(y_i \mid z_i = a\right) = \begin{cases} \sum_{j=1}^{q_1} P\left(m_i = j \mid z_i = 1\right) \cdot f\left(y_i \mid z_i = 1, m_i = j\right), & \text{if } a = 1, \\ \sum_{k=1}^{q_0} P\left(m_i = k \mid z_i = 0\right) \cdot f\left(y_i \mid z_i = 0, m_i = k\right), & \text{if } a = 0, \end{cases}$$

where $P\left(m = j \mid z = 1\right)$ is the mixture weight of a component in class 1, denoted as ξ_{1j} $(j \in \{1, \ldots, q_1\})$; $P\left(m = k \mid z = 0\right)$ is the mixture weight of a component in class 0, denoted as ξ_{0k} $(k \in \{1, \ldots, q_0\})$. These mixture weights satisfy $\sum_{j=1}^{q_1} \xi_{1j} = 1$ and $\sum_{k=1}^{q_0} \xi_{0k} = 1$.

In a Gaussian mixture, it is assumed that each component follows a normal distribution. Hence, in this case we obtain:

$$f\left(y_i \mid z_i = a\right) = \begin{cases} \sum_{j=1}^{q_1} \xi_{1j} \cdot \varphi\left(y_i;\ \mu_{1j}, \sigma_{1j}^2\right), & \text{if } a = 1, \\ \sum_{k=1}^{q_0} \xi_{0k} \cdot \varphi\left(y_i;\ \mu_{0k}, \sigma_{0k}^2\right), & \text{if } a = 0, \end{cases} \tag{2.4}$$

where μ_{1j} is the mean of component j in class 1 and σ_{1j} is its standard deviation; μ_{0k} is the mean of component k in class 0 and σ_{0k} is its standard deviation; $\varphi(y;\ \mu, \sigma^2) = \sigma^{-1}(2\pi)^{-1/2} \exp\left\{-(y-\mu)^2 / (2\sigma^2)\right\}$ denotes the density of a normal distribution with parameters μ and σ^2.

Let $\boldsymbol{\theta} = (\alpha_1, p_{11}, p_{00}, \xi_{11}, \mu_{11}, \sigma_{11}, \ldots, \xi_{1q_1}, \mu_{1q_1}, \sigma_{1q_1}, \xi_{01}, \mu_{01}, \sigma_{01}, \ldots, \xi_{0q_0}, \mu_{0q_0}, \sigma_{0q_0})'$ denote the vector of unknown parameters of the Gaussian mixture model. The identification of all parameters in $\boldsymbol{\theta}$ requires that the order of the components in each class is fixed. For simplicity, we assume here that $\mu_{11} < \ldots < \mu_{1j} < \ldots < \mu_{1q_1}$ and $\mu_{01} < \ldots < \mu_{0k} < \ldots < \mu_{0q_0}$.

In practice, the appropriate numbers of components in the Gaussian mixture models for class 1 and class 0, q_1 and q_0, are not known and need to be determined from the observed data. Here, the main purpose of the mixture model is to provide a flexible way to model the distribution of y in each class. For this type of application, a commonly-used approach is to fit several mixture models to the data with different numbers of components and use the Bayesian information criterion (BIC) to select the optimal number of components (McLachlan and Peel, 2000, page 175 and pages 209-210). However, more recent research suggests that when the components in a mixture model have a very similar mean and variance, the Fisher information matrix may become singular and the BIC is no longer justified as a criterion (Drton and Plummer, 2017).

For such situations, Drton and Plummer (2017) proposed a modified BIC, referred to as the sBIC. In the case study to be discussed in Section 4, we compare the optimal number of components according to the BIC and sBIC criteria.

2.2.2 EM algorithm

Maximum likelihood estimation of the above Gaussian mixture model can be achieved using an EM algorithm (Dempster, Laird and Rubin, 1977; Little and Rubin, 2002). This type of algorithm is often applied when there are unobserved variables in statistical models. As illustrated by its name, it contains two steps: an E step and an M step. The E step builds the expected value of the complete-data log-likelihood function, conditional on the observed data. The M step then estimates the model parameters, which in turn provide input for the next E step. The algorithm proceeds in an iterative way until convergence.

It should be noted that the EM algorithm can estimate all parameters of the mixture model (including the matrix $\mathbf{P}$) from a data set of observations $(\hat{z}_i, y_i)$; thus, it is not necessary to have observed the true class z_i for any unit. Loosely stated, this is possible because, on the one hand, the probability that an observation $(\hat{z}_i, y_i)$ belongs to class 1 or class 0 can be predicted based on differences in the distribution of y for class 1 and class 0 (which is done during the E step) and, on the other hand, the distributions of y and $\hat{z}$ within each true class can be estimated based on these predicted probabilities (which is done during the M step). Technically, estimation is possible because (under normal circumstances) there exists a unique set of parameter values for $\boldsymbol{\theta}$ for which the complete-data log-likelihood function achieves its global maximum. In other words: the model is *identified*; see McLachlan and Peel (2000) for more details.

To derive the complete-data log-likelihood function, we note from expressions (2.2), (2.3), and (2.4) that

$$f(z_i, m_i, \hat{z}_i, y_i; \boldsymbol{\theta}) = \prod_{j=1}^{q_1} \omega_{1ji}^{z_i \mathbf{1}_{(m_i=j)}} \prod_{k=1}^{q_0} \omega_{0ki}^{(1-z_i)\mathbf{1}_{(m_i=k)}},$$

where

$$\omega_{1ji} \triangleq \alpha_1 p_{11}^{\hat{z}_i} (1-p_{11})^{1-\hat{z}_i} \frac{\xi_{1j}}{\sigma_{1j}\sqrt{2\pi}} \exp\left\{-\frac{(y_i-\mu_{1j})^2}{2\sigma_{1j}^2}\right\},$$

$$\omega_{0ki} \triangleq (1-\alpha_1)(1-p_{00})^{\hat{z}_i} p_{00}^{1-\hat{z}_i} \frac{\xi_{0k}}{\sigma_{0k}\sqrt{2\pi}} \exp\left\{-\frac{(y_i-\mu_{0k})^2}{2\sigma_{0k}^2}\right\},$$

and the indicator functions for m_i are defined as:

$$\mathbf{1}_{(m_i=j)} = \begin{cases} 1 & m_i = j \\ 0 & m_i \neq j \end{cases}; \qquad \mathbf{1}_{(m_i=k)} = \begin{cases} 1 & m_i = k, \\ 0 & m_i \neq k \end{cases}.$$

Note that $z_i \sum_{j=1}^{q_1} \mathbf{1}_{(m_i=j)} = z_i$ and $(1-z_i)\sum_{k=1}^{q_0} \mathbf{1}_{(m_i=k)} = 1-z_i$ for all units i.

It follows that the complete-data log-likelihood of the Gaussian mixture model can be written as:

$$\mathrm{LL}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log f(z_i, m_i, \hat{z}_i, y_i; \boldsymbol{\theta}) \;=\; \sum_{i=1}^{N} \left\{ z_i \sum_{j=1}^{q_1} \mathbf{1}_{(m_i=j)} \log \omega_{1ji} + (1-z_i) \sum_{k=1}^{q_0} \mathbf{1}_{(m_i=k)} \log \omega_{0ki} \right\}, \tag{2.5}$$

where we used the assumption that the units are drawn independently of each other.

E step. In the E step of the algorithm, the unobserved quantities $z_i \mathbf{1}_{(m_i=j)}$ and $(1-z_i)\mathbf{1}_{(m_i=k)}$ in (2.5) are replaced by their conditional expectations, given the observed values $\hat{z}_i$ and y_i:

$$\mathrm{E}\left(z_i \mathbf{1}_{(m_i=j)} \,\middle|\, \hat{z}=\hat{z}_i, y=y_i \right) \;=\; P\left(z_i=1, m_i=j \,\middle|\, \hat{z}=\hat{z}_i, y=y_i \right) \;=\; \frac{\omega_{1ji}}{\sum_{j=1}^{q_1} \omega_{1ji} + \sum_{k=1}^{q_0} \omega_{0ki}} \;\triangleq\; A_{1ji};$$

$$\mathrm{E}\left((1-z_i) \mathbf{1}_{(m_i=k)} \,\middle|\, \hat{z}=\hat{z}_i, y=y_i \right) \;=\; P\left(z_i=0, m_i=k \,\middle|\, \hat{z}=\hat{z}_i, y=y_i \right) \;=\; \frac{\omega_{0ki}}{\sum_{j=1}^{q_1} \omega_{1ji} + \sum_{k=1}^{q_0} \omega_{0ki}} \;\triangleq\; A_{0ki}.$$

During iteration t of the algorithm, these expressions are evaluated using the current parameter estimates $\hat{\boldsymbol{\theta}}^{(t-1)}$, yielding the values $A_{1ji}^{(t)}$ and $A_{0ki}^{(t)}$.

M step. In the M step of the algorithm, the log-likelihood function (2.5) is maximised with respect to the model parameters, with the unobserved quantities replaced by $A_{1ji}^{(t)}$ and $A_{0ki}^{(t)}$ from the most recent E step. By setting the first-order partial derivatives of this expected log-likelihood equal to zero, the following formulas are obtained to update the model parameters:

$$\hat{\alpha}_1^{(t+1)} \;=\; \frac{\sum_{i=1}^{N} \left(\sum_{j=1}^{q_1} A_{1ji}^{(t)} \right)}{N};$$

$$\hat{p}_{11}^{(t+1)} \;=\; \frac{\sum_{i=1}^{N} \left(\sum_{j=1}^{q_1} A_{1ji}^{(t)} \right) \hat{z}_i}{\sum_{i=1}^{N} \left(\sum_{j=1}^{q_1} A_{1ji}^{(t)} \right)}; \qquad \hat{p}_{00}^{(t+1)} \;=\; \frac{\sum_{i=1}^{N} \left(\sum_{k=1}^{q_0} A_{0ki}^{(t)} \right) (1-\hat{z}_i)}{\sum_{i=1}^{N} \left(\sum_{k=1}^{q_0} A_{0ki}^{(t)} \right)};$$

$$\hat{\xi}_{1j}^{(t+1)} \;=\; \frac{\sum_{i=1}^{N} A_{1ji}^{(t)}}{\sum_{i=1}^{N} \left(\sum_{j=1}^{q_1} A_{1ji}^{(t)} \right)}; \qquad \hat{\xi}_{0k}^{(t+1)} \;=\; \frac{\sum_{i=1}^{N} A_{0ki}^{(t)}}{\sum_{i=1}^{N} \left(\sum_{k=1}^{q_0} A_{0ki}^{(t)} \right)};$$

$$\hat{\mu}_{1j}^{(t+1)} \;=\; \frac{\sum_{i=1}^{N} A_{1ji}^{(t)} y_i}{\sum_{i=1}^{N} A_{1ji}^{(t)}}; \qquad \hat{\mu}_{0k}^{(t+1)} \;=\; \frac{\sum_{i=1}^{N} A_{0ki}^{(t)} y_i}{\sum_{i=1}^{N} A_{0ki}^{(t)}};$$

$$\hat{\sigma}_{1j}^{(t+1)} \;=\; \sqrt{\frac{\sum_{i=1}^{N} A_{1ji}^{(t)} \left(y_i - \hat{\mu}_{1j}^{(t+1)} \right)^2}{\sum_{i=1}^{N} A_{1ji}^{(t)}}}; \qquad \hat{\sigma}_{0k}^{(t+1)} \;=\; \sqrt{\frac{\sum_{i=1}^{N} A_{0ki}^{(t)} \left(y_i - \hat{\mu}_{0k}^{(t+1)} \right)^2}{\sum_{i=1}^{N} A_{0ki}^{(t)}}}.$$

As noted above, the EM algorithm can be used to estimate the mixture model even when the true class z_i is never observed in the available data. In general, however, the EM algorithm will merely converge to

a local maximum of the log-likelihood function. To ensure that the global maximum is found, it may be necessary to run the algorithm multiple times using different (randomly chosen) starting values and retain the best solution. For general suggestions on how to choose suitable random starting values for mixture models, see McLachlan and Peel (2000, Section 2.12).

Alternatively, observations $(z_i, \hat{z}_i, y_i)$ including the true class may have been obtained for a random subsample of the data (an *audit sample*). If available, an audit sample can be used to improve the convergence of the EM algorithm to the global maximum of the likelihood function, reducing the need for re-runs with different starting values. First, the parameters α_1, p_{11} and p_{00} can be estimated directly from the audit sample to provide reasonable starting values for the EM algorithm. In addition, improved starting values for the other parameters (related to the components of the Gaussian mixture inside each class) could be obtained by applying a k-means clustering algorithm to each true class in the audit sample (Li, 2020b). Finally, for observations from the audit sample, A_{1ji} and A_{0ki} can be replaced during the E step by more narrowly defined expected values:

$$\mathrm{E}\left(z_i \mathbf{1}_{(m_i=j)} \mid z_i=1, \hat{z}=\hat{z}_i, y=y_i\right) = P\left(m_i=j \mid z_i=1, \hat{z}=\hat{z}_i, y=y_i\right) = \frac{\omega_{1ji}}{\sum_{j=1}^{q_1}\omega_{1ji}} \triangleq Q_{1ji};$$

$$\mathrm{E}\left((1-z_i)\mathbf{1}_{(m_i=k)} \mid z_i=0, \hat{z}=\hat{z}_i, y=y_i\right) = P\left(m_i=k \mid z_i=0, \hat{z}=\hat{z}_i, y=y_i\right) = \frac{\omega_{0ki}}{\sum_{k=1}^{q_0}\omega_{0ki}} \triangleq Q_{0ki};$$

whereas $\mathrm{E}\left(z_i \mathbf{1}_{(m_i=j)} \mid z_i=0, \hat{z}=\hat{z}_i, y=y_i\right) = \mathrm{E}\left((1-z_i)\mathbf{1}_{(m_i=k)} \mid z_i=1, \hat{z}=\hat{z}_i, y=y_i\right) = 0.$

2.3 Methods

In this study we will compare three methods that try to estimate the bias and variance of a domain statistic $\hat{\zeta}$ as defined in (2.1): the bootstrap method, the EM bootstrap method and the SIMEX bootstrap method. We did not compare the outcomes with analytical expressions, since we already know that for domain statistics those expressions yield results that are either similar to those of the bootstrap method or less accurate (see Introduction).

As noted at the end of Section 2.1, we are interested here in bias and variance due to classification errors for a finite population, conditional on the values $z_1, \ldots, z_N$ and $y_1, \ldots, y_N$. Due to assumption (2.2), this means that the bias and variance of interest are completely determined by the random process described by the matrix $\mathbf{P}$, applied to the fixed values $z_1, \ldots, z_N$.

An important practical consideration is that the bootstrap and SIMEX bootstrap methods assume that (an estimate of) the matrix $\mathbf{P}$ is available beforehand, whereas an estimate of $\mathbf{P}$ is obtained as part of the EM bootstrap method. For the other methods, $\mathbf{P}$ could be estimated in practice from an audit sample or by running the EM algorithm separately. In our study to be discussed in Sections 3 and 4, we used the estimated $\mathbf{P}$ from the EM algorithm as input for the bootstrap and SIMEX bootstrap methods.

2.3.1 Bootstrap method

The bootstrap method as applied here originates from Van Delden et al. (2016). It is summarised in Algorithm 1. The classification matrix $\mathbf{P}$ is applied to the observed class $\hat{z}$ and bootstrapped classes z^* are obtained. The probability that the bootstrapped class is 1 given the observed class 1 is $P(z_i^* = 1 \mid \hat{z}_i = 1) = p_{11}$ and given the observed class 0 is $P(z_i^* = 1 \mid \hat{z}_i = 0) = 1 - p_{00}$. Through bootstrapping, random classification errors are introduced to observed classes. As a result, estimated bias is computed by comparing bootstrapped statistics ζ^* to the observed statistic $\hat{\zeta}$, and the variance of the bootstrapped statistics ζ^* is an estimate of the variance of the observed domain statistic $\hat{\zeta}$. In practice, the theoretical bootstrap bias and variance are usually approximated using a finite number (S) of bootstrap samples. For certain simple statistics such as α_1 and T_1, it is also possible to derive an exact formula for the theoretical bootstrap bias and variance (Van Delden et al., 2016).

Algorithm 1 The bootstrap method

Input: Observations $(y_i, \hat{z}_i)$ for $i = 1, \ldots, N$, matrix $\mathbf{P}$ and S.

1: **for** $s = 1 \ldots S$ **do**

2: Generate z_i^* by $\mathbf{P}$, conditional on $\hat{z}_i$, for every unit i in the data set

3: Calculate the corresponding ζ^* based on (y_i, z_i^*) instead of $(y_i, \hat{z}_i)$

4: **end for**

5: Calculate $\mathbf{Bias}_{\mathbf{boot}} = \mathrm{E}(\zeta^* \mid \hat{z}) - \hat{\zeta}$, $\mathbf{Var}_{\mathbf{boot}} = \mathrm{Var}(\zeta^* \mid \hat{z})$ based on S simulations

Output: $\mathbf{Bias}_{\mathbf{boot}}$ and $\mathbf{Var}_{\mathbf{boot}}$

It is known that, in general, the bias and variance estimates from Algorithm 1 are biased, due to the fact that the observed classes $\hat{z}_i$ are used as a starting point for the bootstrap. This problem is illustrated in Appendix C.1 using the parameter $\zeta = T_1$, for which an exact analysis is possible. The next two methods attempt to correct for this bias.

2.3.2 EM bootstrap method

In the EM bootstrap method, the observed data $(y, \hat{z})$ are assumed to follow the Gaussian mixture model from Section 2.2. The EM algorithm estimates the parameters $\boldsymbol{\theta}$ of the model, which include the classification probabilities p_{11} and p_{00}. Then we apply a nested simulation process; see Algorithm 2. The purpose of the first level of simulation is to restore (in expectation) an error-free status. It generates classes $\tilde{z}$ from $P(z \mid y, \hat{z}; \hat{\boldsymbol{\theta}})$, leading to unbiased statistics $\tilde{\zeta}$. [Note that the required probabilities are available directly from the EM algorithm, since $P(z_i = 1 \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}}) = \sum_{j=1}^{q_1} A_{1ji}$ and $P(z_i = 0 \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}}) = \sum_{k=1}^{q_0} A_{0ki}$.] After that, classes $\tilde{z}^*$ are generated by a bootstrapping process with matrix $\mathbf{P}$, which leads to bias and variance estimation similar to Algorithm 1. Finally, the results of this inner bootstrap simulation are averaged over the first level of simulation, to reduce the effect of noise due to drawing $\tilde{z}$ from a probability distribution.

Algorithm 2 The EM bootstrap method

Input: Observations $(y_i, \hat{z}_i)$ for $i = 1, \ldots, N$ and S_1, S_2.

Estimate model parameters $\boldsymbol{\theta}$, including p_{11} and p_{00}, from the EM algorithm, conditional on $\hat{z}$ and y

Calculate $P(z_i \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}})$ for every unit i in the data set

for $s_1 = 1, \ldots, S_1$ **do**

  Generate $\tilde{z}_i$ by $P(z_i \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}})$ for every unit i in the data set

  Calculate the corresponding $\tilde{\zeta}$ based on $(y_i, \tilde{z}_i)$ instead of $(y_i, \hat{z}_i)$

  for $s_2 = 1, \ldots, S_2$ **do**

    Generate $\tilde{z}_i^*$ by $\mathbf{P}$, conditional on $\tilde{z}_i$, for every unit i in the data set

    Calculate the corresponding $\tilde{\zeta}^*$ based on $(y_i, \tilde{z}_i^*)$ instead of $(y_i, \hat{z}_i)$

  end for

end for

Calculate $\mathbf{Bias}_{\mathbf{comb}} = \mathrm{E}_{\tilde{z}}(\mathrm{E}(\tilde{\zeta}^* \mid \hat{z}, \tilde{z}) - \tilde{\zeta} \mid \hat{z})$ and $\mathbf{Var}_{\mathbf{comb}} = \mathrm{E}_{\tilde{z}}(\mathrm{Var}(\tilde{\zeta}^* \mid \hat{z}, \tilde{z}) \mid \hat{z})$ based on S_1 and S_2 simulations

Output: $\mathbf{Bias}_{\mathbf{comb}}$, $\mathbf{Var}_{\mathbf{comb}}$ and estimated matrix $\mathbf{P}$

It is shown in Appendix C.2 that, for $S_1, S_2 \to \infty$, Algorithm 2 indeed yields approximately unbiased bias and variance estimators in the special cases $\zeta = T_1$ and $\zeta = \alpha_1$, provided that the observed data follow the assumed mixture model. For other, non-linear parameters such as $\zeta = \sigma_1$, no exact proof is available, but we will investigate the behaviour of Algorithm 2 in a simulation study.

2.3.3 SIMEX bootstrap method

SIMEX was introduced by Cook and Stefanski (1994) for numerical variables. It uses a bootstrapping process to add various extra errors, through which a sequence of estimates under different error-included conditions is obtained. Then a function is applied to extrapolate the sequence back to the estimate without error. Here, we use an extension of the SIMEX method to categorical variables that was introduced by Küchenhoff et al. (2006).

Traditionally, the SIMEX method would be applied to obtain a bias-corrected estimate of the target parameter itself (ζ in our notation). Here, we use it to obtain bias-corrected bootstrap estimates of the bias and variance of $\hat{\zeta}$. We will refer to this approach as the SIMEX bootstrap method.

The SIMEX bootstrap method simulates multiple conditions where classification errors are added to the data according to the matrix $\mathbf{P}^{\lambda}$ [with $\mathbf{P}$ given by (2.3)], for different values of $\lambda \geq 0$. For each value of λ, the bootstrap method of Algorithm 1 is applied to the adjusted data to obtain bias and variance estimates. Finally, the SIMEX bias and variance estimates are obtained by extrapolating the sequence of bias and variance estimates as functions of λ to the value $\lambda = -1$. This can be understood as follows. The available observed data can be viewed as one realisation of applying the matrix $\mathbf{P}$ to the true data. So, starting from the observed data at $\lambda = 0$ we want to extrapolate the sequence of bias and variance estimates back to what

would have been found at the unobserved point of zero misclassifications, which corresponds to $\lambda = -1$. The SIMEX bootstrap method is summarised in Algorithm 3.

Algorithm 3 The SIMEX bootstrap method

Input: Observations $(y_i, \hat{z}_i)$ for $i = 1, \ldots, N$, matrix $\mathbf{P}$ and S_1, S_2.

1: **for** λ ranging from 0 to 5 with increments of 0.5 **do**
2: Calculate $\mathbf{P}^{\lambda}$
3: **for** $s_1 = 1, \ldots, S_1$ **do**
4: Generate $z_i^{(\lambda)}$ by $\mathbf{P}^{\lambda}$, conditional on $\hat{z}_i$, for every unit i in the data set
5: Calculate the corresponding $\zeta^{(\lambda)}$ based on $(y_i, z_i^{(\lambda)})$ instead of $(y_i, \hat{z}_i)$
6: **for** $s_2 = 1, \ldots, S_2$ **do**
7: Generate $\hat{z}_i^{(\lambda)}$ by $\mathbf{P}$, conditional on $z_i^{(\lambda)}$, for every unit i in the data set
8: Calculate the corresponding $\hat{\zeta}^{(\lambda)}$ based on $(y_i, \hat{z}_i^{(\lambda)})$ instead of $(y_i, \hat{z}_i)$
9: **end for**
10: **end for**
11: Calculate $\mathbf{Bias}_{\mathbf{simex}}^{(\boldsymbol{\lambda})} = \mathrm{E}_{z^{(\lambda)}}(\mathrm{E}(\hat{\zeta}^{(\lambda)} \mid \hat{z}, z^{(\lambda)}) - \zeta^{(\lambda)} \mid \hat{z})$ and $\mathbf{Var}_{\mathbf{simex}}^{(\boldsymbol{\lambda})} = \mathrm{E}_{z^{(\lambda)}}(\mathrm{Var}(\hat{\zeta}^{(\lambda)} \mid \hat{z}, z^{(\lambda)}) \mid \hat{z})$ based on S_1 and S_2 simulations
12: **end for**
13: Extrapolate $\mathbf{Bias}_{\mathbf{simex}}^{(\boldsymbol{\lambda})}$ and $\mathbf{Var}_{\mathbf{simex}}^{(\boldsymbol{\lambda})}$ to $\lambda = -1$ to get $\mathbf{Bias}_{\mathbf{simex}}$ and $\mathbf{Var}_{\mathbf{simex}}$

Output: $\mathbf{Bias}_{\mathbf{simex}}$, $\mathbf{Var}_{\mathbf{simex}}$

Calculate $\mathbf{P}^{\boldsymbol{\lambda}}$. When $\lambda = 0$, $\mathbf{P}^0$ is an identity matrix. In this special case, no additional classification errors are introduced in line 4 of the SIMEX bootstrap algorithm. For any real-valued $\lambda > 0$, $\mathbf{P}^{\lambda}$ can be computed using the eigenvalue decomposition of $\mathbf{P}$. Through it, we get $\mathbf{P} = \mathbf{QAQ}^{-1}$, where $\mathbf{A}$ is a diagonal matrix of eigenvalues and $\mathbf{Q}$ is a matrix of eigenvectors of $\mathbf{P}$. Then $\mathbf{P}^{\lambda}$ is calculated by $\mathbf{P}^{\lambda} = \mathbf{QA}^{\lambda}\mathbf{Q}^{-1}$. The corresponding probabilities $p_{11}^{(\lambda)}$ and $p_{00}^{(\lambda)}$ are obtained:

$$\mathbf{P}^{\lambda} = \begin{pmatrix} p_{11}^{(\lambda)} & 1 - p_{11}^{(\lambda)} \\ 1 - p_{00}^{(\lambda)} & p_{00}^{(\lambda)} \end{pmatrix},$$

where $p_{11}^{(\lambda)}$ is the probability of $z^{(\lambda)} = 1$ when $\hat{z} = 1$, and $p_{00}^{(\lambda)}$ is the probability of $z^{(\lambda)} = 0$ when $\hat{z} = 0$. These probabilities are used to draw $z_i^{(\lambda)}$, given $\hat{z}_i$, in line 4 of the algorithm. Note that this procedure works only if $\mathbf{P}^{\lambda}$ is a true probability matrix in the sense that $0 \le p_{11}^{(\lambda)}, p_{00}^{(\lambda)} \le 1$. For $\lambda \ge 0$, this is guaranteed provided that $p_{11} + p_{00} > 1$ (Küchenhoff et al., 2006). For $\lambda < 0$ this property does not hold; hence, extrapolation is a necessary step of the SIMEX method.

Extrapolation functions. A SIMEX method yields a consistent estimator of a parameter of interest (in our case: bias and variance of $\hat{\zeta}$) when the extrapolation function, which describes how the uncorrected estimator varies as a function of λ, is correctly specified; see Küchenhoff et al. (2006) for more details. In

the literature on SIMEX, extrapolation functions that have been suggested include: local linear regression (LOESS) on λ (Hopkins and King, 2010) and standard regression on a quadratic or cubic polynomial of λ (Küchenhoff et al., 2006). We have compared all three approaches in the simulation study of Section 3.

3. Simulation study

3.1 Settings

We simulated a population of size $N = 2{,}000$ with two classes. For target variable y we used the following Gaussian mixture distribution: Class 0 has one component with $\mu_0 = 15$, $\sigma_0 = 3$; class 1 has two components with $(\xi_{11}, \xi_{12}) = (0.5, 0.5)$, $(\mu_{11}, \mu_{12}) = (2, 4)$, and $(\sigma_{11}, \sigma_{12}) = (1, 2)$.

With respect to the true classification variable z, we tested two different proportions of class 1: α_1 is 0.3 or 0.5. The observed classification variable $\hat{z}$ was generated from z by using the transition matrix $\mathbf{P}$ given in equation (2.3). In the simulation study, values of p_{11} and p_{00} were set at 0.6, 0.75, or 0.9. For each setting of α_1, p_{11} and p_{00}, we used $S_0 = 100$ implying that 100 sets of $\hat{z}$ were generated from z. In each set s_0, 5% of the units from the population (so 100 in total) were randomly selected as an audit sample which was used to obtain the starting values for the EM algorithm of $\hat{p}_{11}$, $\hat{p}_{00}$ and the other class-level parameters. The starting values of the component-level parameters were obtained by k-means; see Li (2020b) for more details. In a preliminary study, we have also tested the EM algorithm without an audit sample (see Section 3.3).

For each set s_0, the bootstrap method, the SIMEX bootstrap method and the EM bootstrap method were applied to estimate the corresponding bias and variance of the estimated domain parameters $\hat{\zeta}$ that are given in Table 2.1: the total sum for class 1 (T_1), the proportion of class 1 (α_1) and the standard deviation for class 1 (σ_1). The bias and variance estimates are given here as the average over the S_0 sets.

With respect to the SIMEX bootstrap method, we tested the above settings of the simulation study for three different extrapolation functions, as noted in Section 2.3.3; see also Appendix A. The estimates of the bias and of the standard error of the LOESS function and of the third order polynomial were closer to the true values than those of the second order polynomial. Since the LOESS function was used in a previous study on misclassifications (see Hopkins and King (2010)) we used this extrapolation function in the remainder of this paper.

The EM algorithm was stopped either when none of the parameter estimates changed by more than 0.001 between two iterations, or after 5,000 iterations. In practice, both in this simulation study and in the case study of Section 4, this maximum number of iterations was never reached. For about 1% of the simulated data sets, the EM algorithm did not converge properly due to numerical issues. These were caused by an unfortunate choice of starting values estimated from the audit sample, for instance a starting value of p_{11} exactly equal to 1. In principle, this problem could be avoided easily by a slight change of starting values. However, as this issue only affected a small number of cases, for convenience we ignored these cases in the results below.

The number of iterations in the methods, S, S_1 and S_2 were all set at 100. To obtain a benchmark for the true bias and variance of the estimated domain parameters $\hat{\zeta}$, 1,000 sets of $\hat{z}$ were simulated.

3.2 Results

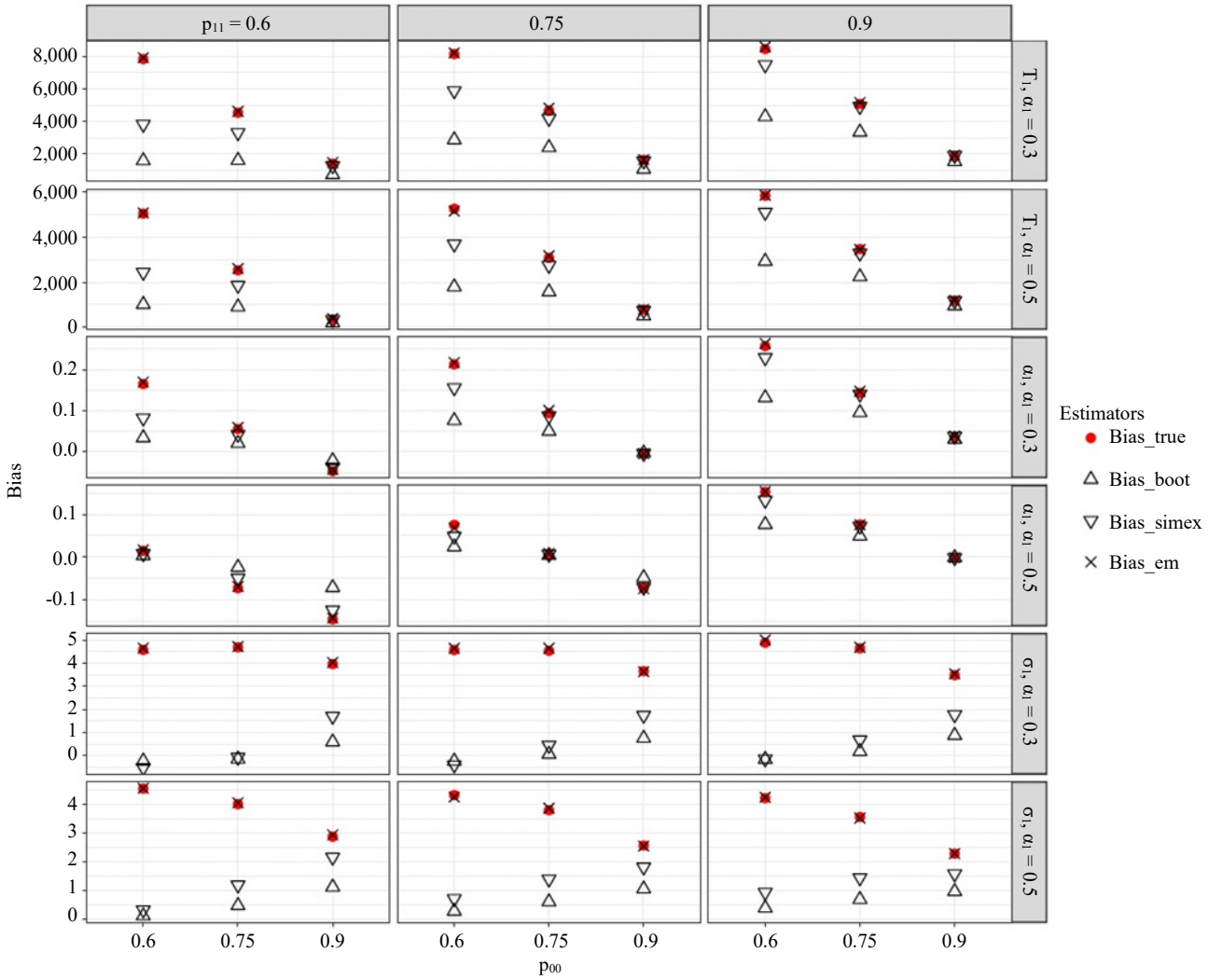
Bias estimation. Figure 3.1 shows the bias of $\hat{T}_1$, $\hat{\alpha}_1$ and $\hat{\sigma}_1$ for $\alpha_1 = 0.3$ (row 1, 3 and 5 respectively) and for $\alpha_1 = 0.5$ (row 2, 4 and 6 respectively). Each column shows results under the three p_{11} settings. In each subplot, the horizontal axis indicates values of p_{00}, the vertical axis defines the estimates under different conditions and methods. Furthermore, the different methods are given by different symbols: the true values (•), the bootstrap (Δ), the SIMEX bootstrap (∇) and the EM bootstrap method ($\times$). Figure 3.2 shows the standard error (square root of the variance) of $\hat{T}_1$, $\hat{\alpha}_1$ and $\hat{\sigma}_1$ for $\alpha_1 = 0.3$ and $\alpha_1 = 0.5$ for the same settings.

Overall, for a given value of p_{11}, the bias of $\hat{T}_1$, $\hat{\alpha}_1$ and $\hat{\sigma}_1$ decreased with a larger value of p_{00}. For a given value of p_{00}, the bias increased with a larger value of p_{11}. This result can be understood as follows. In the case of the total, an analytical expression for the bias of $\hat{T}_1$ is given by (C.1). Normally, this expression cannot directly be computed since T_0 and T_1 are unknown in real situations, but in the simulations we know their values. In our example, at $\alpha_1 = 0.3$ one finds that $T_0 = 15 \times (1 - 0.3) \times 2{,}000 = 21{,}000$ and $T_1 = (0.5 \times 2 + 0.5 \times 4) \times (0.3) \times 2{,}000 = 1{,}800$. An increase of p_{00} from 0.6 to 0.9 for a given value of p_{11} reduces the bias since the contribution $(1 - p_{00})\, T_0$ drops from 8,400 to 2,100. This refers to units with true class 0 that are erroneously observed as class 1 (overestimation). Conversely, an increase of p_{11} from 0.6 to 0.9 for a given value of p_{00} leads to a small increase of the bias due to an increase in the contribution $(p_{11} - 1)\, T_1$ from -720 to -180. This contribution refers to units with true class 1 that erroneously have an observed class 0 (underestimation). Generally, for a given value of p_{11} the bias of the statistics of interest ($\hat{T}_1$, $\hat{\alpha}_1$ and $\hat{\sigma}_1$) decreases with a larger value of p_{00} because their overestimation decreases. Conversely, for a given value of p_{00} the bias of the statistics increases with a larger value of p_{11} because its underestimation decreases.

With respect to the three estimation methods (see Figure 3.1), we found that the bias estimates for the statistics of interest from the EM bootstrap method were closest to the true values. The bias estimates from the SIMEX method were closer to the true values than the estimates from the bootstrap method, but they still had a considerable distance to the true bias. Furthermore, we found that the bias estimates from all three methods were closer to their true bias for T_1 and σ_1 when the misclassification probabilities were reduced (p_{11} or p_{00} values closer to 1). When p_{11} and p_{00} were equal to 0.9, bias estimates from the bootstrap method and the SIMEX bootstrap method even overlapped with the corresponding true bias. Finally note that when $\alpha_1 = 0.5$ and $p_{00} = p_{11}$, then the true bias of $\hat{\alpha}_1$ equals 0 and all three methods estimated this bias correctly.

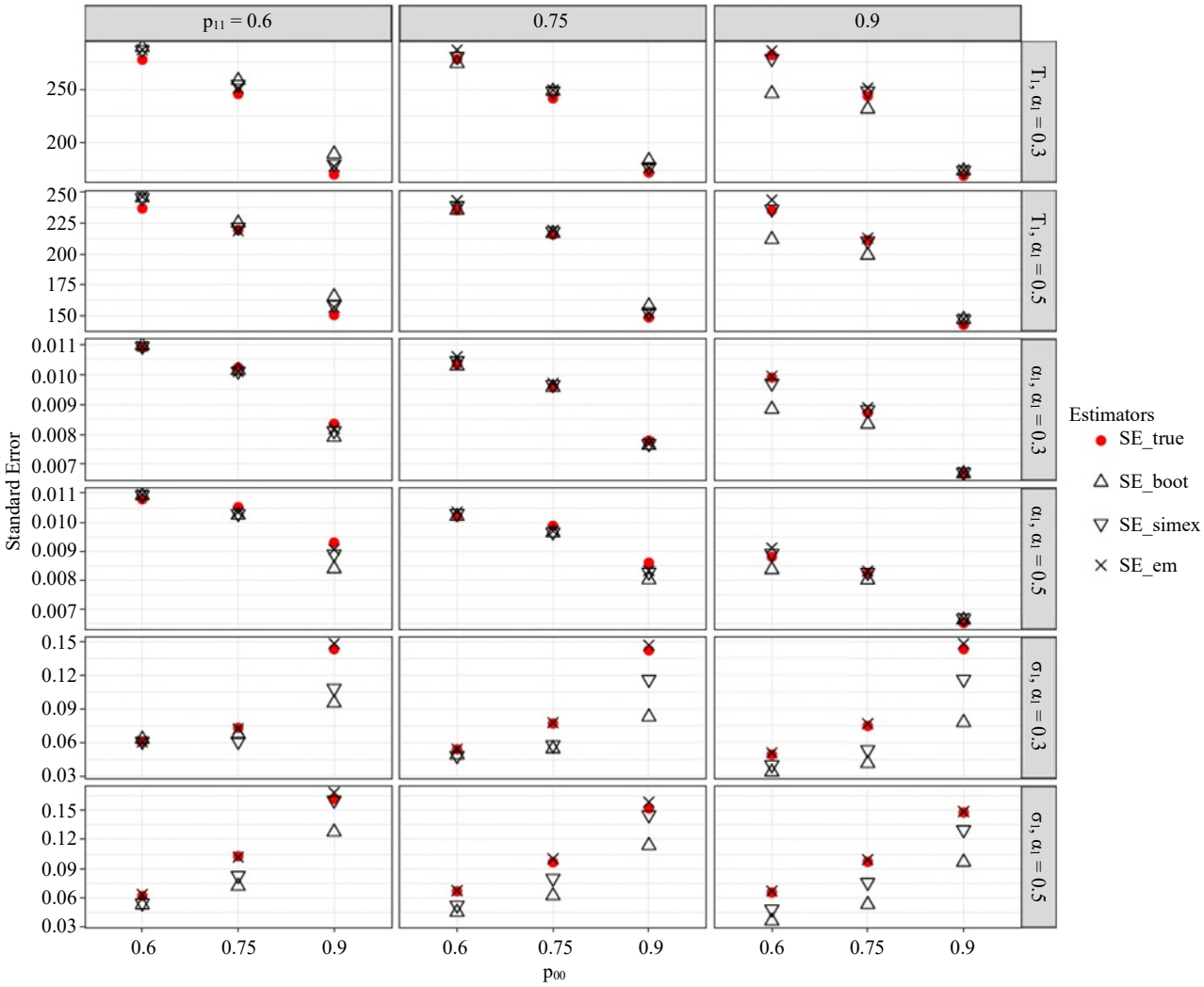
Variance estimation. Figure 3.2 shows the true and estimated standard error of $\hat{T}_1$, $\hat{\alpha}_1$ and $\hat{\sigma}_1$ for $\alpha_1 = 0.3$ (row 1, 3 and 5 respectively) and for $\alpha_1 = 0.5$ (row 2, 4 and 6 respectively). The true standard errors of $\hat{T}_1$ and of $\hat{\alpha}_1$ were reduced with less misclassifications with p_{00} and p_{11} going from 0.6 to 0.9. This result follows from the analytical expression (C.2) for the variance of $\hat{T}_1$.

Figure 3.1 Bias estimation in the simulation study.



Notes: Rows represent different domain parameters $(T_1, \alpha_1, \sigma_1)$ for two settings of α_1: $\alpha_1 = 0.3$ and $\alpha_1 = 0.5$. Columns represent different p_{11} values, within each column different p_{00} values are given.

Figure 3.2 Standard error estimation in the simulation study.



Notes: For row and column settings see Figure 3.1.

Surprisingly, the standard error of $\hat{\sigma}_1$ clearly increased for a given value of p_{11} when p_{00} increased from 0.6 to 0.9. Furthermore, there was a very small reduction of this standard error for a given value of p_{00} when p_{11} increased from 0.6 to 0.9. This result can be explained as follows. The mean turnover level is much higher in class 0 than in class 1. When most units observed in class 1 truly come from class 1 and relatively few units from class 0 ($p_{00} = 0.9$), the standard error of $\hat{\sigma}_1$ is large since there is a considerable variation over the replicates in which turnover values of class 0 will be observed as class 1. In some of the replicates this concerns outlying values compared to the true distribution in class 1. When the number of units from class 0 increases ($p_{00} = 0.75$), this variation in turnover values of the replicates decreases and thus the standard error of $\hat{\sigma}_1$ decreases. Since turnover values of class 0 are larger than those of class 1, the impact of varying values of p_{00} is larger than for p_{11}.

With respect to the three estimation methods (see Figure 3.2) we found, similar as with the bias, that the standard error estimates for the statistics of interest from the EM bootstrap method were closest to the true values. For $\hat{T}_1$ and $\hat{\alpha}_1$, the standard error estimates from all three methods were almost equally good in most conditions. In the other conditions, the bootstrap method estimates were least accurate, followed by those of the SIMEX bootstrap method which were close to the true values. The estimated standard errors of $\hat{\sigma}_1$ were close to their true values in the case of the EM bootstrap method, much closer than for the other two methods. Larger values of the standard error of $\hat{\sigma}_1$ lead to larger estimation differences among the three methods.

3.3 Additional simulation studies

Below we summarise the results of three additional simulation studies.

Audit sample. In a preliminary study we have compared the estimation of bias and standard error with and without an audit sample (Li, 2020a). We tested p_{00} and p_{11} values of 0.6, 0.75 and 0.9, N = 2,000, α_1 equal to 0.1, 0.3, 0.5, 0.7 and 0.9; a single Gaussian component in each class, with μ_1 values of 2, 10, 12 and 15, μ_0 fixed at 15, $\sigma_1 = 1$ and $\sigma_0 = 2$. In situations without an audit sample, different starting values were tested for the EM algorithm. For p_{00} and p_{11}, starting values of 0.6, 0.75 and 0.9 were used, leading to nine combinations for the pair (p_{00}, p_{11}). By choosing different values for p_{00} and p_{11}, the starting points can be seen as representative in the parameter space. The starting value of α_1 was set according to $\alpha_1 = \sum \hat{z}_i / N \times (3 - p_{11} - p_{00}) - (1 - p_{00})$ which is an unbiased estimate of α_1 (Kloos et al., 2021). Means and variances were started by robust statistics obtained from observed classes, where μ_g for the two classes was initialised at the median of the corresponding target variable, and σ_g started with $k \times$ MAD where $k = 1/\Phi^{-1}(3/4) \approx 1.48$ and MAD is the median absolute deviation; see Rousseeuw and Croux (1993).

We found that the estimated bias of the statistics of interest using the EM bootstrap with and without an audit sample yielded nearly the same results, except for difficult estimation conditions. These difficult conditions were when $\mu_0 = \mu_1$ combined with lower values for α_1 and smaller p_{00} and p_{11} values (not shown). Under those conditions the bias estimates were less accurate, but the true (relative) bias was small. The bias estimates were close to zero while the true relative bias was up to 0.03. For the standard error of the statistics of interest, the EM bootstrap with and without an audit sample yielded nearly the same results under all tested conditions.

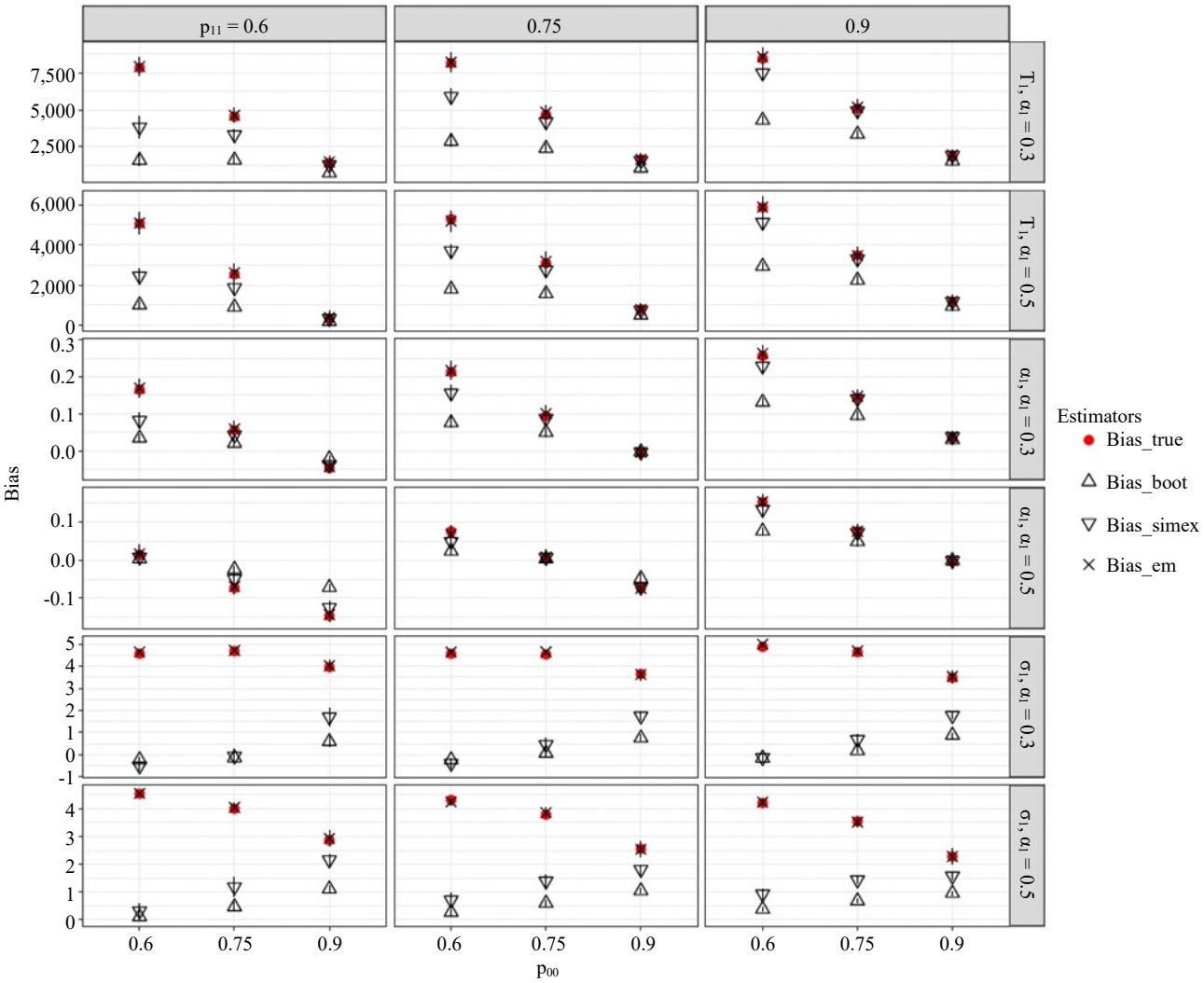
Results for μ_1. Besides the estimated bias and standard error of $\hat{T}_1$, $\hat{\alpha}_1$ and $\hat{\sigma}_1$ for α_1 = 0.3, we have also computed them for $\hat{\mu}_1$. Note that $\hat{\mu}_1 = \hat{T}_1 / (\hat{\alpha}_1 \times N)$, so results for $\hat{\mu}_1$ follow from those of $\hat{T}_1$ and $\hat{\alpha}_1$. With respect to the comparison of the three methods, we concluded that the true bias and standard error were estimated most accurately by the EM bootstrap method. The specific results can be found in Li (2020a).

Confidence intervals. So far, we have presented the results as averaged over S_0 = 100 sets of $\hat{z}$ that were generated from z. In a practical situation one would have only a single set of $\hat{z}$ values. This raises the question whether the EM bootstrap method also leads to more accurate bias estimates than the other two

methods in the case of a single sample or only on average. To that end, we estimated a 95% confidence interval for the bias estimates of the statistics of interest for the three methods using the $S_0 = 100$ replicates. This interval was estimated as 1.96 times the standard error of the estimated bias which in turn was derived from the $S_0 = 100$ bias estimates.

From Section 3.2 we have already concluded that in some settings all three methods yielded accurate bias estimates for $\hat{T}_1$ and $\hat{\alpha}_1$, but otherwise the bias estimate by the EM bootstrap method was clearly closer to the true value than that of the SIMEX bootstrap and bootstrap method, as averaged over S_0 replicates. Since the 95% confidence intervals were very small for all three methods (see Figure 3.3), in most settings also for a single set of $\hat{z}$ values, the EM bootstrap estimate of the bias was closer to the true bias than the estimates from the SIMEX bootstrap and bootstrap method.

Figure 3.3 Bias estimation in the simulation study with 95% confidence interval.



Notes: For row and column settings see in Figure 3.1.

4. Case study

4.1 Data

In order to assess the performance of our methods in real applications, a case study was conducted. In the case study, the bias and variance of the estimated domain statistics $\hat{T}_1$, $\hat{\alpha}_1$ and $\hat{\sigma}_1$ were estimated, using the same methods as in the simulation study.

For the case study, we started with a data set that contains the logarithm of the yearly turnover for a population of enterprises for which we know their website address; see Oosterveen (2020). The enterprises are classified by economic activity codes, according to the European NACE rev. 2 classification (Eurostat, 2008). The enterprises, their NACE codes and the website addresses were obtained from a statistical business register (SBR). For some enterprises we obtained the website address from a data set with URLs that we retrieved from an external company DataProvider.

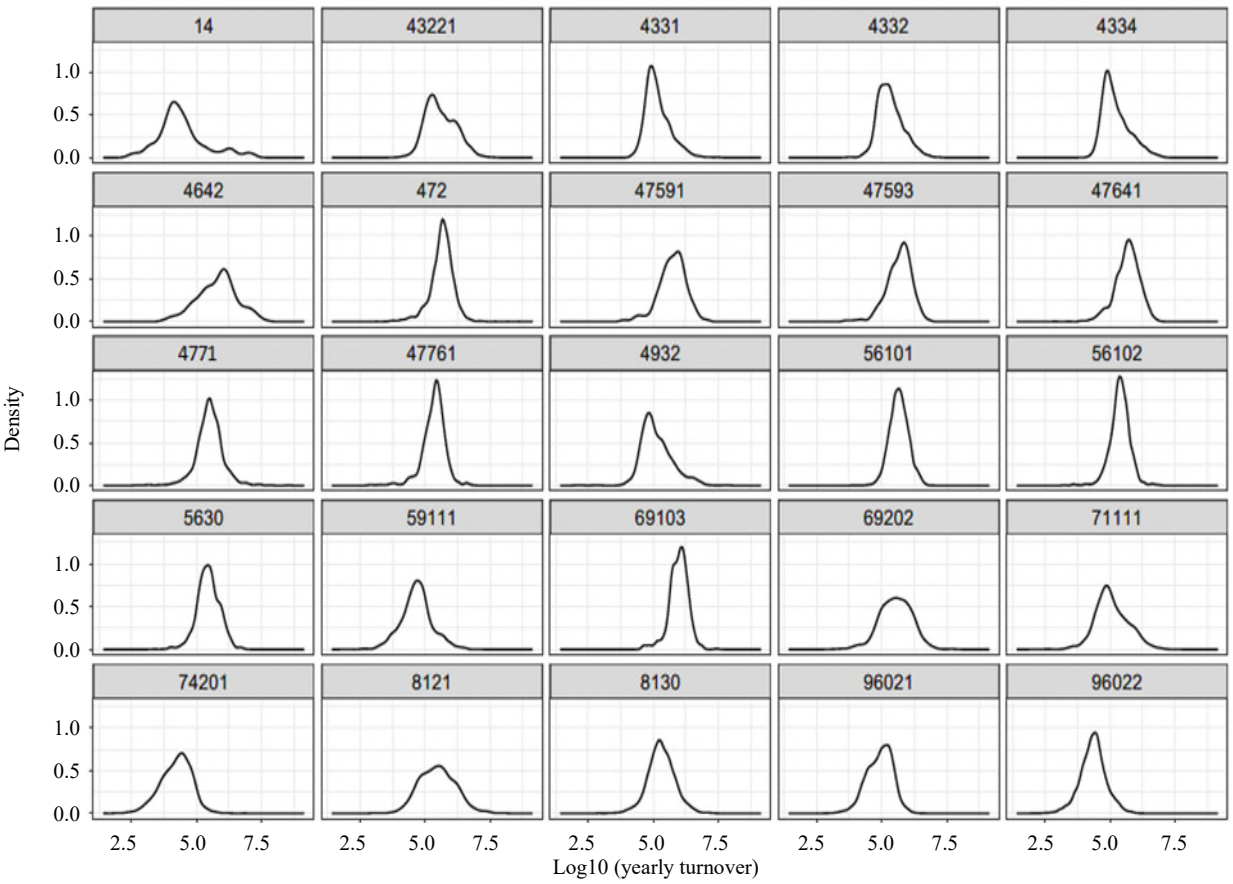
These NACE code values are prone to classification errors. NACE codes of larger and more complex enterprises are checked manually and corrected if needed. For the case study we therefore limited ourselves to the simpler enterprises, that are composed of three legal units or fewer, since those are the enterprises whose NACE codes are the most prone to misclassification in practice. Furthermore, we started with a shortlist of 25 economic activities; the NACE codes are given in Figure 4.1. This shortlist contains a few groups of NACE codes with similar classes within a group, such as wholesale of clothes and retail trade of clothes, and dissimilar to others, such as taxi operation.

Similar to the simulation study, in the case study we wanted to start with a data set that was free of misclassifications, and then introduce the misclassifications on purpose in order to test the performance of the three methods. However, the data that we extracted from the SBR concerned observed NACE codes that could already contain misclassifications. We therefore did not use all smaller enterprises with a website (76,270 enterprises), but we used a selection of enterprises that had a relatively high probability of having the correct NACE code. This selection was made by predicting the NACE code of the enterprises based on the text of the main page of the website of the enterprise. These texts were scraped and preprocessed; see Oosterveen (2020) on how this was done. Three different machine learning algorithms (Naïve Bayes, Support Vector Machine and Random Forest) were trained to predict the NACE code, using a ten fold cross-validation procedure. In each fold, the model was trained on 90% of the data and the fitted model was used to predict the remaining 10% of the data. An observed NACE code was considered to be correct when the fitted models of all three algorithms predicted the same code or when it was predicted by two of the algorithms and the prediction confidence of the models was relatively high; see Oosterveen (2020) for a more detailed description of the selection. This selection led to 45,965 enterprises.

In the present paper we limit ourselves to binary misclassifications. We therefore made a further selection of pairs of two NACE codes out of the shortlist of 25 NACE codes. We selected different pairs, where each selected pair is referred to as a "case". The pairs differed in the extent of overlap between the log-turnover distributions and in the shapes of the distributions. Furthermore, we ensured that the two group sizes were

not too small and not too unbalanced, which meant that α_1 was not close to 0 or 1. The full set of tests that we did can be found in Li (2020b). Here we present only the results for the two most interesting pairs which are given in Table 4.1.

Figure 4.1 Density distribution of log turnover for all groups.



Notes: The labels refer to NACE codes. NACE = Nomenclature générale des activités économiques dans les Communautés européennes.

Table 4.2 gives some basic statistics of each of the four subpopulations defined by the NACE codes selected in Table 4.1: the size, total, mean and standard deviation of the log yearly turnover per enterprise. Case 1 has two well-separated distributions and a large number of enterprises per class. By contrast, in case 2 the two distributions are less well-separated and they have a smaller number of enterprises per class. Table 4.2 describes the true domain statistics ζ of each class.

Before we applied our methods, we removed outliers. We removed enterprises with a turnover value that was below $Q_1 - 1.5 \times \text{IQR}$ or above $Q_3 + 1.5 \times \text{IQR}$, where Q_1 is the first quartile of the enterprises in the true class, Q_3 is the third quartile and $\text{IQR} = Q_3 - Q_1$. Removal of outliers was only necessary in case 2. We removed 17 outliers from NACE 4932 and 6 from NACE 8121.

Table 4.1
The selected groups and their allocations in the case study.

NACE Code	Description of Economic Activity	Case	Class
56101	Restaurants	Case 1	Class 1
96022	Beauty treatment, pedicures and manicures, make-up and image consulting	Case 1	Class 0
4932	Taxi operation	Case 2	Class 1
8121	General cleaning of buildings	Case 2	Class 0

Note: NACE = Nomenclature générale des activités économiques dans les Communautés européennes.

4.2 Settings

In contrast to the simulation study, here the number of components in the Gaussian mixture model is not known, and needs to be determined. To that end, we fitted a standard Gausian mixture model (i.e. without a misclassification component) for each class per case separately. We then determined the BIC and sBIC of these fitted models, since those measures could be used to select the number of components, see Section 2.2. For convenience, we fitted this Gausian mixture model to the true data rather than to the generated data with misclassifications, thereby avoiding that we had to run it for 100 (S_0) sets times nine misclassification conditions (see below). In practice, one has to fit the standard Gaussian mixture model to observed data containing misclassifications, which is expected to result in slightly more components than when the model is run on true data. In Appendix B, the estimated optimal number of components for case 1 and 2 are shown according to the BIC and sBIC criteria. The optimal number of components in class 0 of case 2 was two using sBIC and one using BIC. For the other three classes no differences in the optimal number of components were found. For the remainder, we used the number of components according to the sBIC criterion, shown in Table 4.2.

Similar to the set up of the simulation study (Section 3.1), the observed classification variable $\hat{z}$ was generated from z by using $\mathbf{P}$ (equation 2.3) with values of p_{11} and p_{00} of 0.6, 0.75, or 0.9. For each setting of p_{11} and p_{00}, we generated $S_0 = 100$ sets of $\hat{z}$ from z. In each set s_0, 5% units from the population were randomly selected as the audit sample, which was used to estimate $\hat{p}_{11}$ and $\hat{p}_{00}$. These estimates were used as starting values for the EM algorithm. The final estimates of $\hat{p}_{11}$ and $\hat{p}_{00}$ by the EM algorithm were input for the bootstrap and SIMEX bootstrap methods. The number of iterations S, S_1 and S_2 was 100. The overall bias and variance estimates of the three methods were computed as the average over S_0 bias and variance estimates. As before, the estimated true bias and variance of the estimated domain parameters were based on 1,000 sets of $\hat{z}$.

Table 4.2
Domain statistics for each class in the case study.

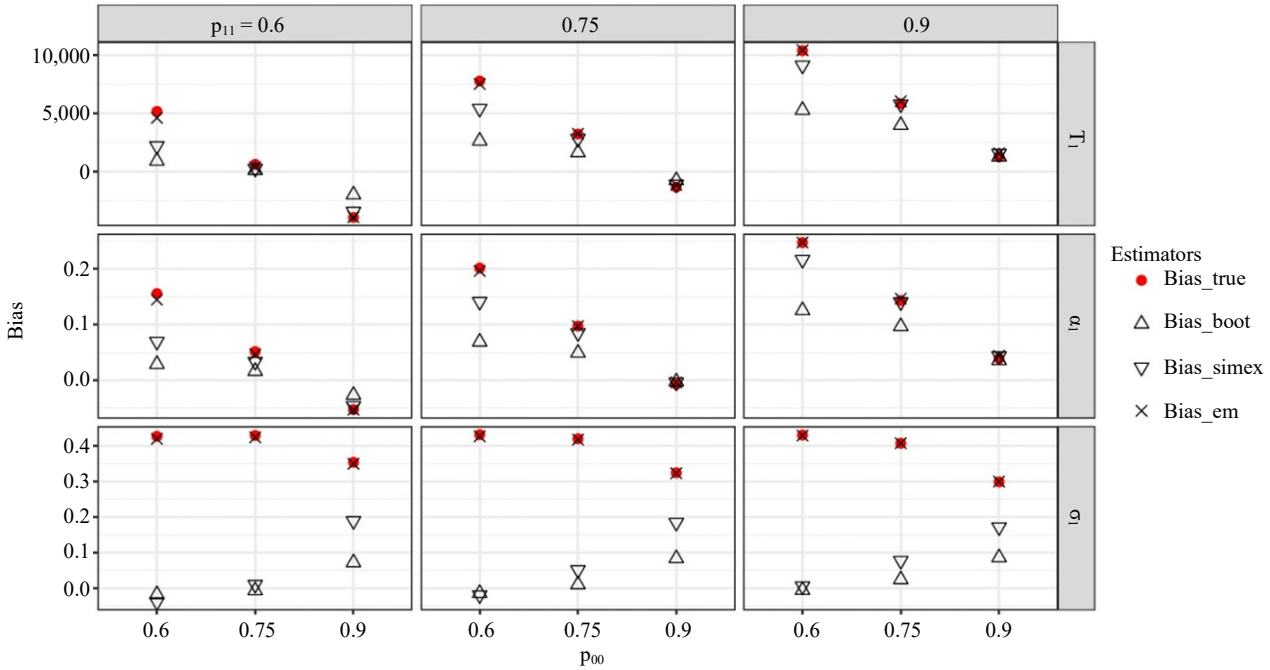
Case	Class	Number of Components	Size	Total	Mean	Standard Deviation
Case 1	Class 1	1	3,076	17,415	5.66	0.358
	Class 0	2	6,993	30,377	4.34	0.501
Case 2	Class 1	2	642	3,294	5.13	0.597
	Class 0	2	1,067	5,847	5.48	0.687

4.3 Results

The pattern of the bias (see Figures 4.2 and 4.3) and of the standard error (Figures 4.4 and 4.5) as a function of p_{00} and p_{11} is the same as has been found previously in the simulation study. For most settings tested, the estimation of the bias of $\hat{T}_1$, $\hat{\alpha}_1$ and $\hat{\sigma}_1$ was most accurate by the EM bootstrap method, followed by the SIMEX bootstrap method, while the bootstrap led to the least accurate results. An exception occurred in case 2, $p_{00} = 0.9$ for $\hat{\sigma}_1$ where the SIMEX bootstrap method was most accurate followed by the EM bootstrap method. In some of the settings with $p_{00} = 0.75$ the EM bootstrap method and the SIMEX bootstrap method yielded near-identical results for the bias.

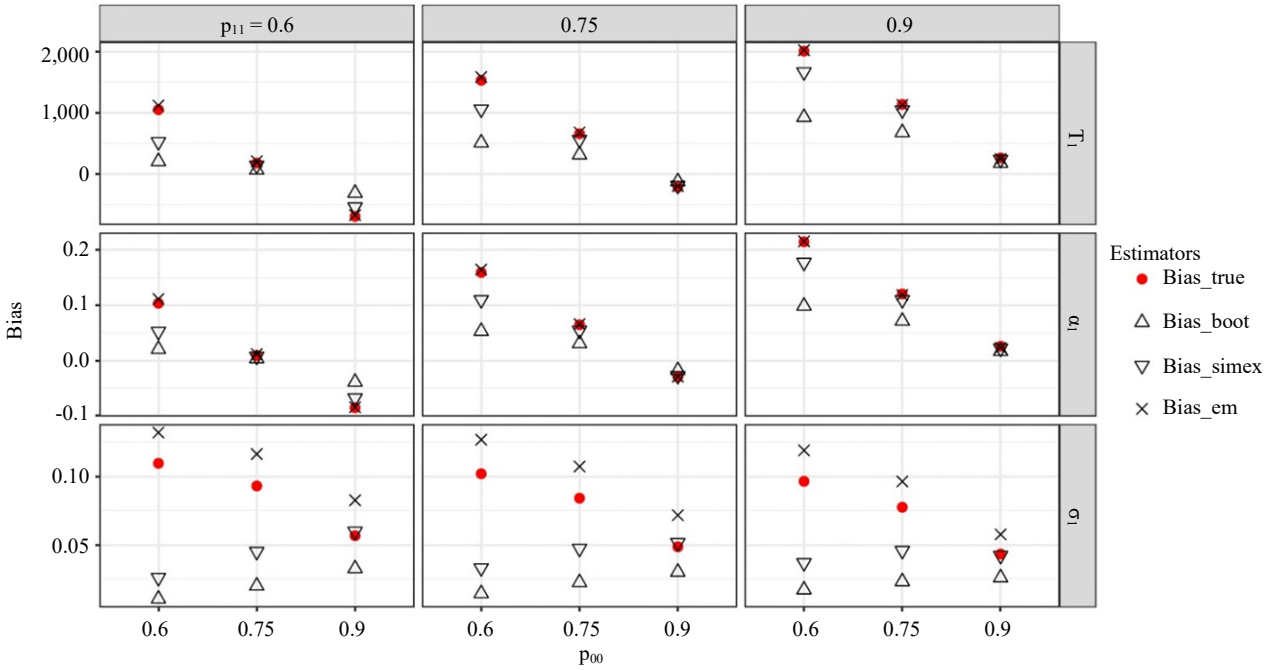
The bias estimates by the EM bootstrap method almost overlapped with the corresponding true values. Only in case 2, the bias estimates of $\hat{\sigma}_1$ showed some distance from the true values, while the bias for $\hat{T}_1$ and $\hat{\alpha}_1$ remained accurate. It is seen in Figure 4.1 that the true distribution of y in class 1 in this case (NACE code 4932) is more right-skewed than the other distributions considered in this study. This may particularly affect the statistic $\hat{\sigma}_1$, which is relatively sensitive to values in the right tail of the distribution. A follow-up analysis showed that the EM bootstrap method performed somewhat better in this example when the number of components per class was increased from two to three (see Figures B.1 and B.2 in Appendix B). This suggests that it may be beneficial to choose a relatively large number of mixture components if the distribution is known to be asymmetrical and if non-linear, non-robust parameters such as σ_1 are of interest. Note, however, that the statistic $\hat{\sigma}_1$ is not directly published as output in official statistics. Overall, the results suggest that the EM bootstrap method usually has accurate performance even in difficult situations, when the two classes contain fewer units and their distributions overlap.

Figure 4.2 Bias estimation in case 1.



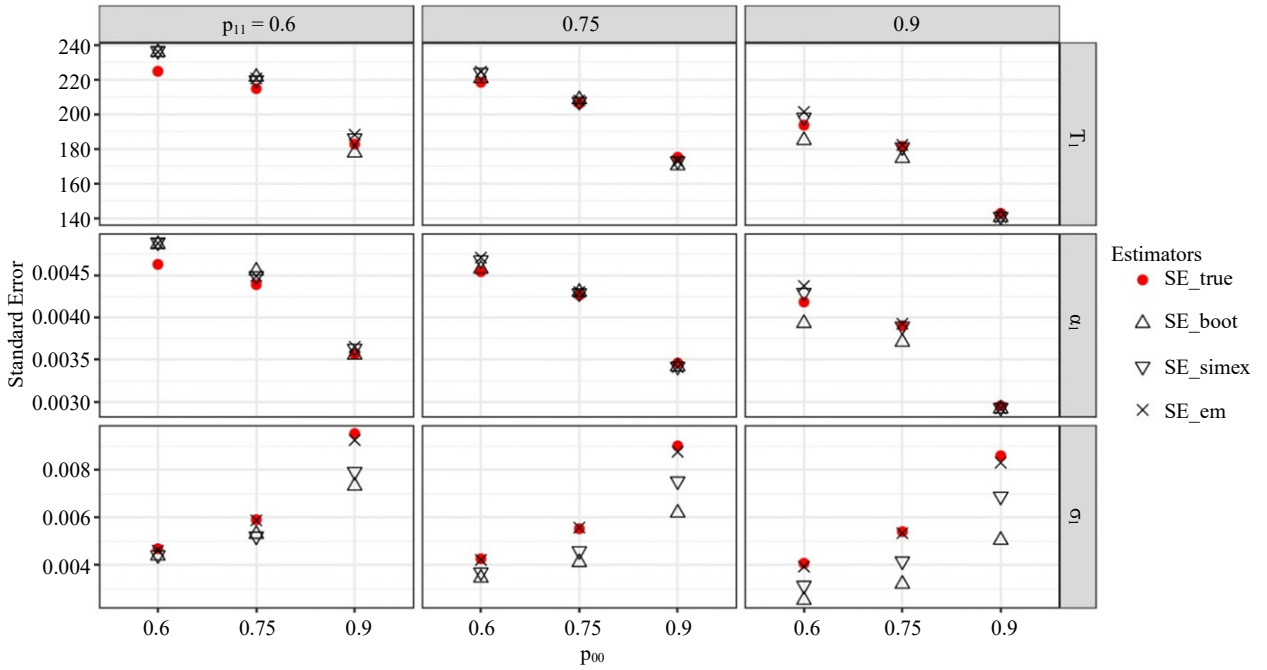
Notes: Rows represent different domain parameters (T_1, α_1, σ_1). Columns represent different p_{11} values, within each column different p_{00} values are given.

Figure 4.3 Bias estimation in case 2.



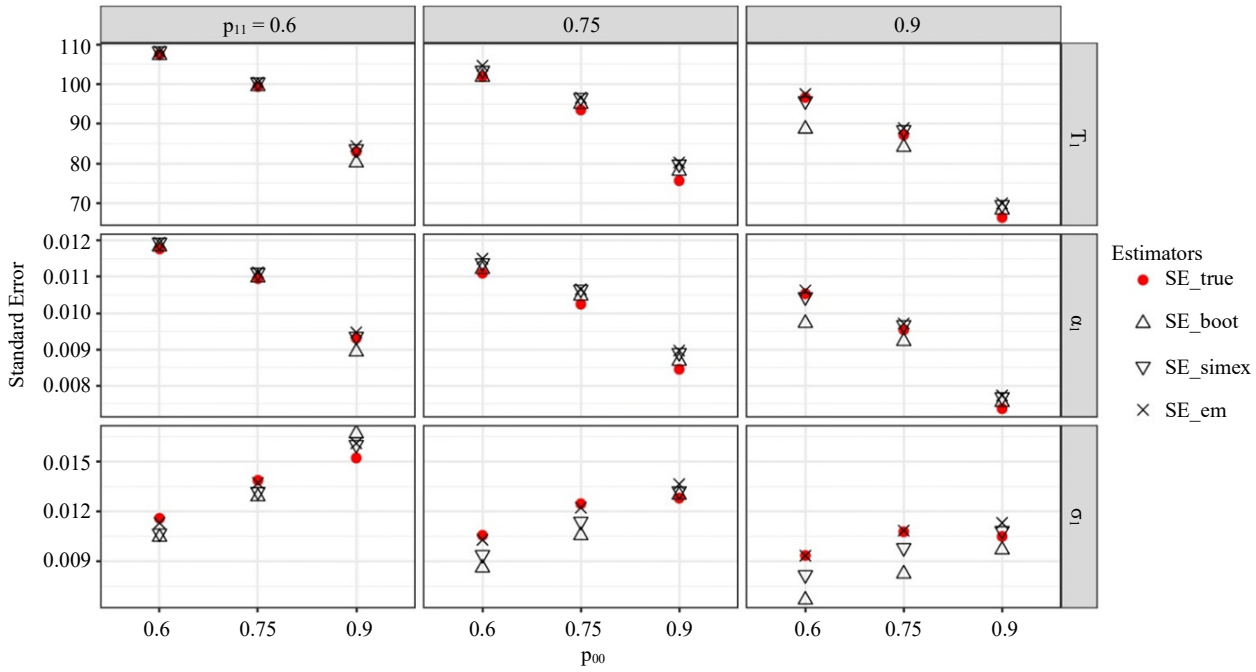
Notes: For row and column settings see Figure 4.2.

Figure 4.4 Standard error estimation in case 1.



Notes: For row and column settings see Figure 4.2.

Figure 4.5 Standard error estimation in case 2.



Notes: For row and column settings see Figure 4.2.

Differences among the three methods with respect to the accuracy of the estimated standard error of $\hat{T}_1$, $\hat{\alpha}_1$ and $\hat{\sigma}_1$ were smaller than for the estimated bias. In settings where the estimates of the three methods were clearly different, most of the times the standard error estimate by the EM bootstrap method was closest to its true value, followed by the SIMEX bootstrap while the bootstrap method was the least accurate. Sometimes the estimated standard error of the SIMEX bootstrap was closest to the true standard error, for instance for case 1, $p_{00} = 0.6$, $p_{11} = 0.9$ and target statistic $\hat{T}_1, \hat{\alpha}_1$ but differences with the EM bootstrap method were small.

5. Discussion

In this paper, we have proposed an EM bootstrap method for estimating the accuracy of domain statistics, in terms of bias and variance, in the presence of misclassifications. The use of an EM algorithm is not new in studies where classification errors occur. For instance, Sinclair and Hooker (2017) and Kosinski and Flanders (1999) both aim to estimate parameters of a model in the presence of classification errors in one or more of its variables. Our EM bootstrap method assumes a Gaussian mixture model. In our study, we do not use the mixture model to directly improve the accuracy of our (total or mean) estimates, but instead we use it to estimate the accuracy of a given estimator. The main reason is that National Statistical Institutes tend to avoid using model-based estimators directly for output, particularly if assumptions of the model are not verifiable (Van den Brakel and Bethlehem, 2008). However, using models to estimate the quality of output is accepted. An additional reason why we used the method to estimate the accuracy rather than to (directly) improve the accuracy is that the performance needed for the former is less than for the latter. Once

the bias and variance of the (given) estimates are obtained, one either concludes that those estimates are sufficiently accurate and publishes them, or one aims to first improve the accuracy of the estimates. In the latter case one might apply data editing first to reduce the rate of misclassifications. Alternatively, the mixture model could then be used to construct an improved estimate, if the model is considered sufficiently trustworthy.

We compared how well one can estimate the bias and variance of statistics in the presence of misclassification with the EM bootstrap method, the bootstrap method and the SIMEX bootstrap method using simulated data sets and real applications. The simulated data concerned (mixtures of) normal distributions whereas the real data concerned empirical distributions. For most of the conditions tested, we found that the EM bootstrap method outperformed the bootstrap and the SIMEX bootstrap method. The estimated bias of statistics from the EM bootstrap method was closer to the true bias than for the bootstrap and the SIMEX bootstrap method. The estimated variance of statistics based on the EM bootstrap method was also more accurate than for the other methods, but here the relative differences between the estimated values of the three methods were smaller than for the bias. Only in situations where the means of the two distributions were very close together, and populations were small the $\hat{\sigma}_1$ statistic was better estimated with SIMEX, but still the bias estimate of the total remained accurate.

Based on the results obtained we expect that the EM bootstrap outperforms the bootstrap for a binary variable as long as the model parameters are well estimated and the distribution of y is captured well. We found that the EM algorithm only had difficulties in estimating the model parameters well when the two distributions had means that were close together while the population size was small. Still, under those difficult conditions the bias of the total was estimated well. We expect that the combination of similar class means and large standard deviations is also more problematic. The SIMEX method only works well when the parameter of interest is a smooth function of λ that can be extrapolated well. Since our estimation method does not depend on these conditions we expect it to perform better than SIMEX for binary classification variables. When the distribution of y is very skewed and contains a number of outliers, then the SIMEX algorithm might outperform the EM algorithm.

The above results were obtained by averaging over 100 simulation runs. By comparing the 95% confidence intervals of the bias estimated in the simulation study, we showed that even in a practical situation where there is only *one single set* of $\hat{z}$, the EM bootstrap method leads to better bias estimates than the bootstrap method and the SIMEX bootstrap method. We obtained these results, by using the classification error probabilities as estimated by the EM algorithm as input for the bootstrap and SIMEX bootstrap. That way, we gave the bootstrap and SIMEX bootstrap the best possible starting position to compete with the EM bootstrap. As an alternative, we also used estimated probabilities from an audit sample as input to the bootstrap and SIMEX bootstrap methods, which resulted in poorer results than before (not shown here).

Besides giving more accurate bias and variance estimates, the EM bootstrap method has two further advantages over the bootstrap and the SIMEX bootstrap method. The first advantage is that the EM bootstrap does not require that classification error probabilities p_{11} and p_{00} are accurately estimated beforehand; in fact, the EM algorithm provides estimates of these probabilities as part of its output. In previous studies these misclassification probabilities could only be obtained from audit samples (Van Delden et al., 2016).

The results of Li (2020a) and Li (2020b) suggest that the EM bootstrap also works well without an audit sample, except in some extreme cases (where two classes had the same mean). When there is no audit sample available, one should use multiple starting values to avoid finding a local maximum. If an audit sample is available, it is useful to incorporate it into the EM algorithm (Li, 2020a). In our study, the starting values obtained from the audit sample were sufficient to obtain accurate bias and variance estimates. The second advantage of the EM bootstrap method is that its output includes unit-specific probabilities $P(z_i = 1 \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}}) = \sum_{j=1}^{q_1} A_{1ji}$ and $P(z_i = 0 \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}}) = \sum_{k=1}^{q_0} A_{0ki}$, which could be used to predict the probability of a classification error in $\hat{z}_i$ for each unit in the data set. That outcome could subsequently be used to manually check and correct units with a potentially incorrect code, in an efficient way. This might save considerable time and effort compared to simply checking all units in a (sub)population based on p_{11} and p_{00}.

There are two points of attention with respect to the practical use of the EM bootstrap method. First, the EM bootstrap method assumes that the actual distribution can be described with a Gaussian mixture model. According to McLachlan and Peel (2000) a Gaussian mixture model can accommodate various distributions for the continuous variable. In the case studies we found that the empirical distributions could already be approximated with 2-3 components. We do not expect that the Gaussian mixture assumption will pose great limits to its application in practice, although a caveat should be made that we did not test our method in situations that require more than three Gaussian components per class. Furthermore, in our case studies the number of components per class was determined using sBIC on the true data, whereas in practice it would have to be determined on data with misclassifications. The results in Section 4.3 suggest that, for skewed data, it could be beneficial in practice to err on the side of including too many components rather than too few. These matters could be investigated further.

Second, the number of iterations of the two loops will need to be determined with care since it will influence the accuracy of the bias and variance estimates of the EM bootstrap method. (This similarly holds for the other two methods.) In our study, we used a fixed number of iterations for the experiments. Based on the results, we judged that we had performed enough iterations to draw valid conclusions. In practice though, the number of iterations should be adjusted according to properties of the data sets, such as size, distribution of each class, distance between the two classes, etc. Efron and Tibshirani (1993) provide some theoretical considerations to take into account when choosing the number of iterations in a bootstrap algorithm; for instance, more iterations are usually needed for a smaller population size (op. cit., Section 6.4). In addition, it has been shown for similar nested algorithms that the number of iterations in the outer for-loop has the strongest effect on convergence (Chang and Hall, 2015), which suggests that increasing S_1 in Algorithm 2 is more beneficial than increasing S_2. However, as each application is different, it is good practice to examine the convergence of bootstrap estimates, e.g., by plotting intermediate results against the number of iterations to check when they become sufficiently stable. Finally, when one is only interested to estimate the bias, and not the variance, then the inner bootstrap loop within the EM bootstrap method is not needed, which saves computation time; see the "EM method" in Li (2020b).

In a future study, a number of extensions of our approach would be useful. A first, important, extension would be to generalise the bias and variance estimation to a situation with a classification variable with $D \geq 2$ classes. With D classes, the probabilities of misclassification will be given by a $D \times D$ matrix $\mathbf{P}$

with $D(D-1)$ probabilities that are to be estimated. For each additional domain $d \in \{1, \ldots, D\}$, the number of other parameters in the mixture model increases linearly according to $(q_d - 1) + 2q_d + 1 = 3q_d$ where q_d is the number of components per additional domain d. Because the number of parameters increases with the number of domains the audit sample becomes more important to give the model reasonable starting values. It is also important to test whether such a D class EM bootstrap converges well. Furthermore, it needs to be tested whether the mixture model performs better than the existing SIMEX and bootstrap methods for $D > 2$.

A second extension would be to account for sample data rather than census data. In that case output quality is affected by both classification error and sampling error. In the case of simple random sampling the estimated bias will not be affected by the sampling error. For the variance one could use two approaches. One approach is that the sampling procedure is also bootstrapped by including it in the EM bootstrap procedure. Alternatively, a hybrid approach could be used in which the EM bootstrap estimate is used for the classification errors while an analytical expression is used for the sampling error.

A third extension would be to relax the assumptions for the probabilities of classification errors. In our study, the probabilities of making classification errors, p_{11} and p_{00}, were assumed independent of the continuous variable. However, in real situations, this assumption does not always hold, at least not without conditioning on other covariates. It would therefore be interesting to make an extension in which the misclassification probabilities depend on covariates.

A fourth extension would be to use multiple numerical target variables, such as height and weight of patients in medical records. Then there will be more than one target variable in the general model. A multivariate Gaussian mixture model can be a suitable model for this case (McLachlan and Peel, 2000). A fifth possible extension is to take missing values in the target variable(s) into account.

Finally, we note that Algorithm 1 in our study is a standard, single bootstrap method. In the literature, double and higher-order bootstraps have also been proposed as a way of correcting for bias in the bias and variance estimators from a single bootstrap (Chang and Hall, 2015; Hall and Martin, 1988). These methods do not require an explicit model for the data but, like the single bootstrap, they do require (an estimate of) the matrix $\mathbf{P}$ as input. It would be interesting to compare the performance of the EM bootstrap method and a double bootstrap in a future study, in particular for the extended problem with $D \gg 2$ classes and/or several numerical target variables, for which estimating a Gaussian mixture model may become challenging.

Acknowledgements

The views expressed in this article are those of the authors and do not necessarily reflect the policies of Statistics Netherlands. The authors would like to thank the Associate Editor and two anonymous referees for their constructive and detailed comments on a draft version of this article.

Appendix

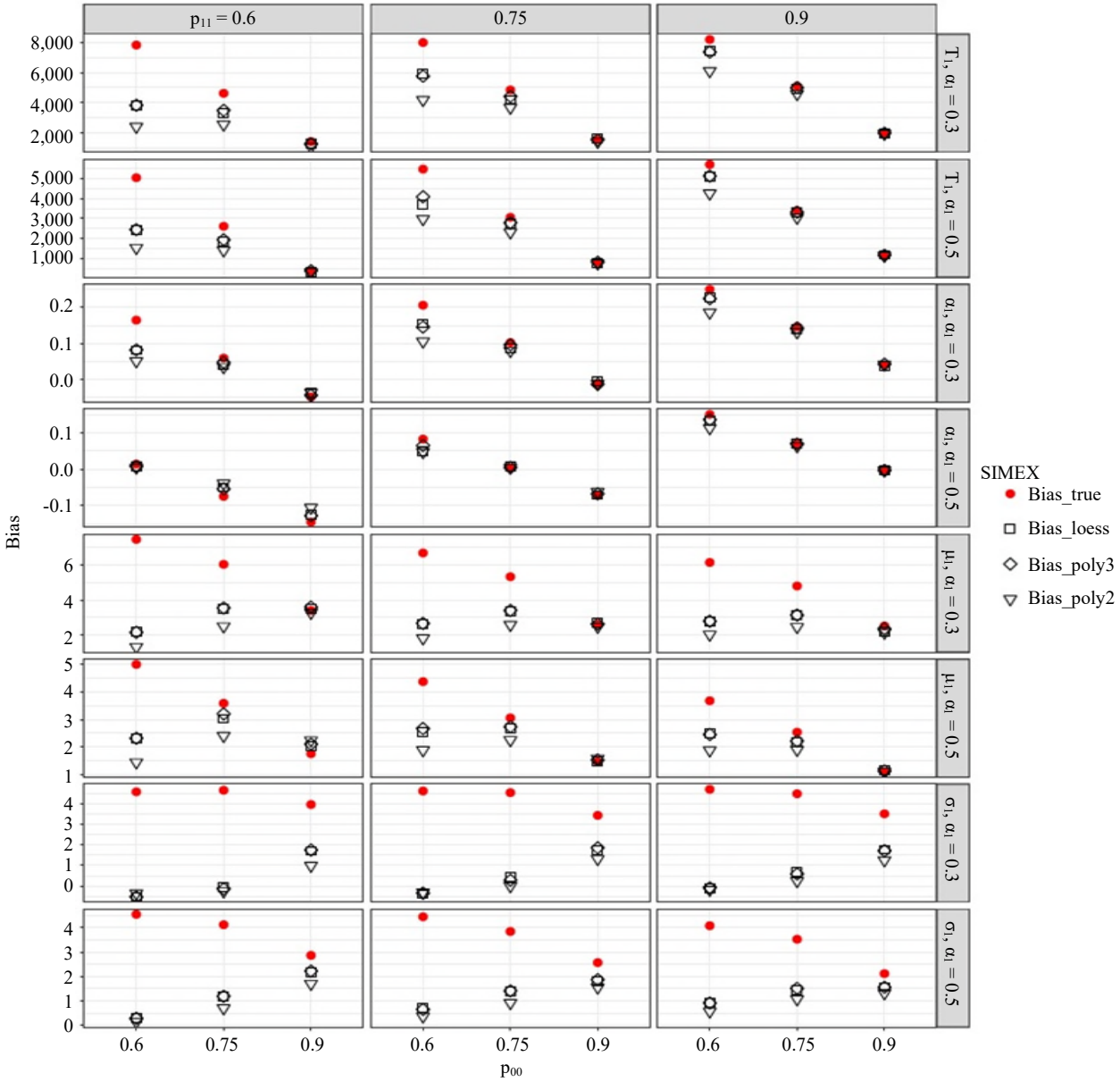
A. Extrapolation function in the SIMEX bootstrap method

In this appendix, three extrapolation functions used in the SIMEX bootstrap method are compared: a local polynomial regression (LOESS), a second order polynomial regression (poly2) and a third order

polynomial regression (poly3). For the LOESS function we used default settings of the loess function in R (span = 0.75 and nls.control(maxiter = 1000)).

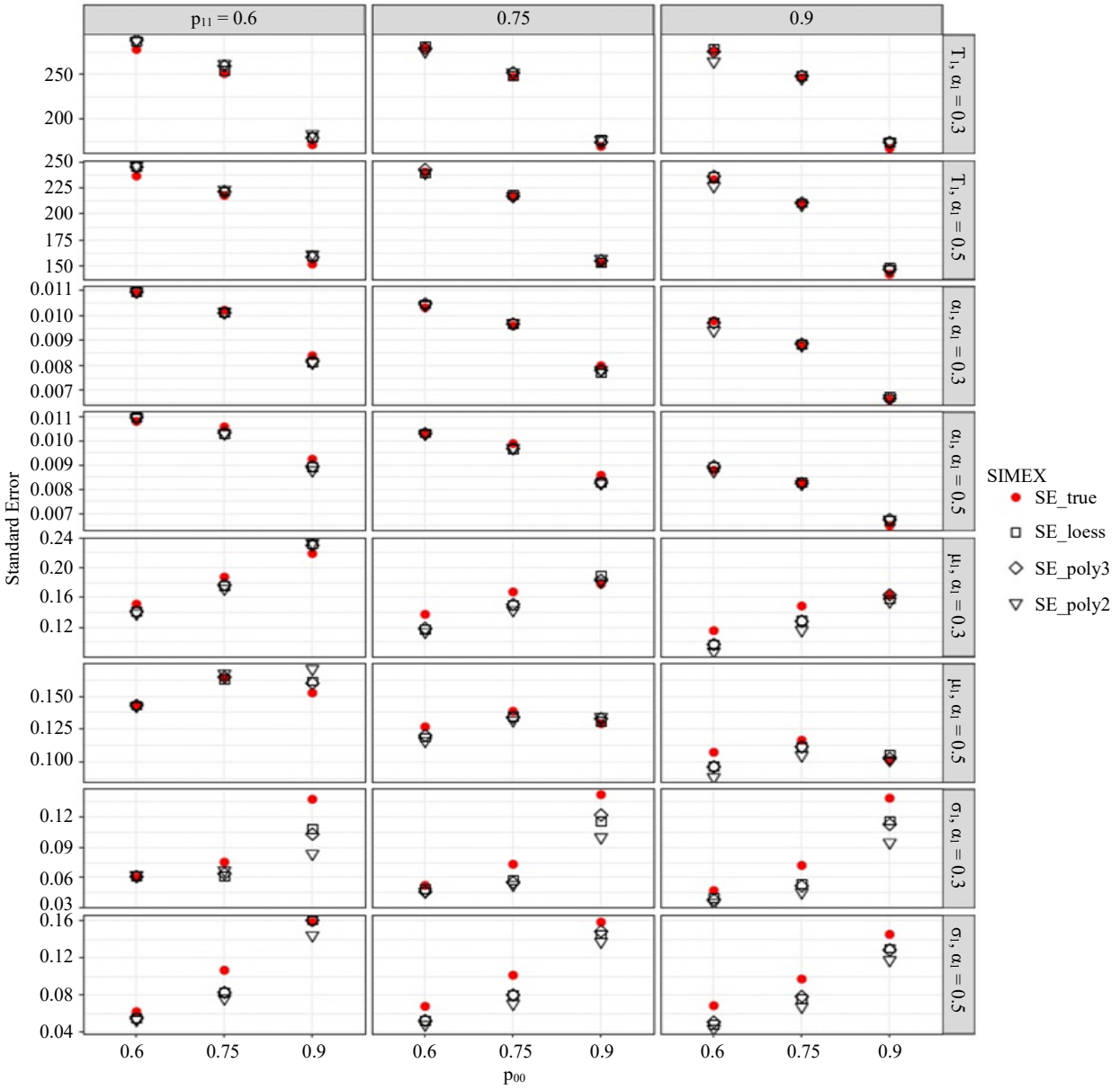
The bias and standard error estimates from the LOESS function and of the third order polynomial regression were similar, see Figures A.1 and A.2. The estimates of both models were closer to the true values than those of the second order polynomial regression. Considering that the LOESS function had been used in a previous study on misclassifications (Hopkins and King, 2010), we decided to apply the LOESS function in the present paper.

Figure A.1 Comparison of the bias estimation performance of the three extrapolation functions in the SIMEX bootstrap method.



Notes: Rows represent different domain parameters $(T_1,\ \alpha_1,\ \mu_1,\ \sigma_1)$ for two settings of α_1: $\alpha_1 = 0.3$ and $\alpha_1 = 0.5$. Columns represent different p_{11} values, within each column different p_{00} values are given.

Figure A.2 Comparison of the standard error estimation performance of the three extrapolation functions in the SIMEX bootstrap method.



Notes: For row and column settings see Figure A.1.

B. BIC vs. sBIC

Before fitting the Gaussian mixture model, one must select its number of components. In this appendix, we compare the bias and the standard error estimation outcomes when using the BIC versus the sBIC as criterion to select the number of components. The sBIC is more justified when the components of the mixture model have very similar means and variance; see Section 2.2.1.

Table B.1 shows the optimal number of components selected by the BIC and sBIC criteria. The two criteria led to the same number of components for both classes of case 1 and for class 1 of case 2. For class 0 of case 2, the optimal number of components selected by sBIC was two and by BIC it was one. Hence,

only for case 2, we compared the bias and the standard error estimates, using two (sBIC) or one (BIC) component for class 0.

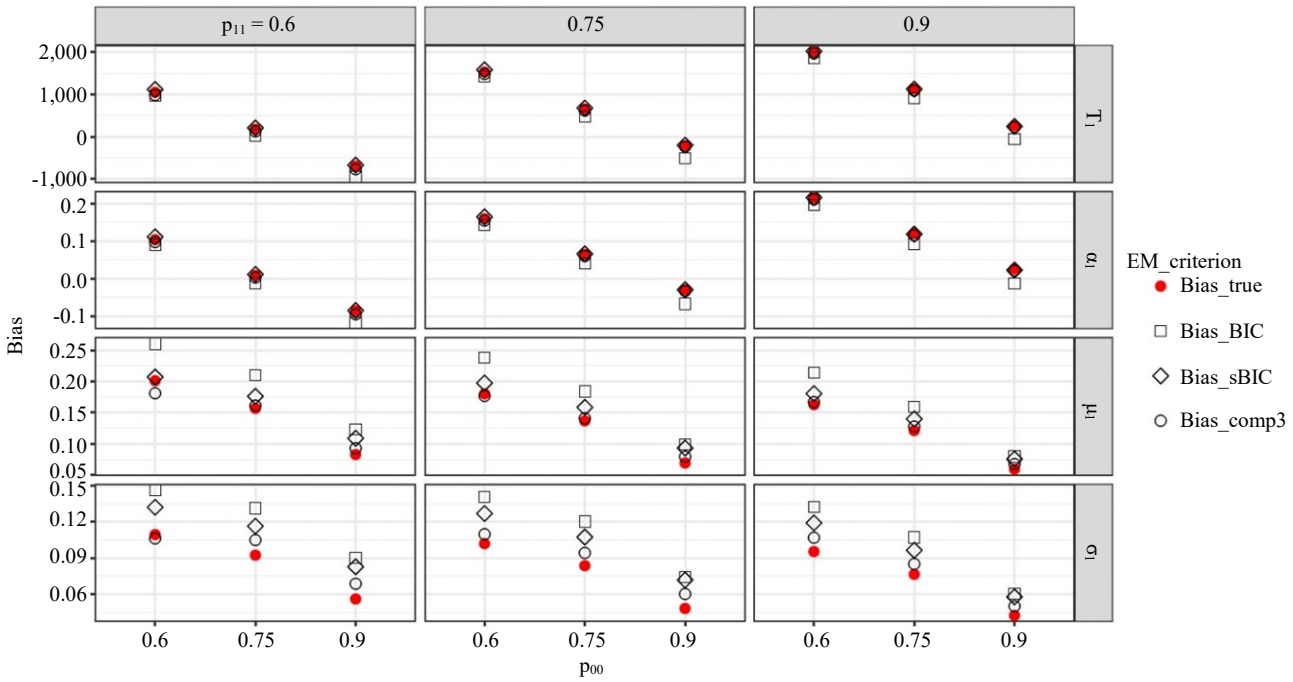
Table B.1
The optimal number of components selected by BIC and sBIC.

Case No.	Class	Optimal Number	
		BIC	sBIC
Case 1	Class 1	1	1
	Class 0	2	2
Case 2	Class 1	2	2
	Class 0	1	2

Using two components for class 0 led to more accurate bias estimates than using one component; see Figure B.1. For some of the settings the standard error estimates based on two components were also more accurate than those based on one component, see Figure B.2, although the accuracy differences were smaller for the standard error than for the bias. We therefore decided to use the sBIC in our case study to estimate the optimal number of components.

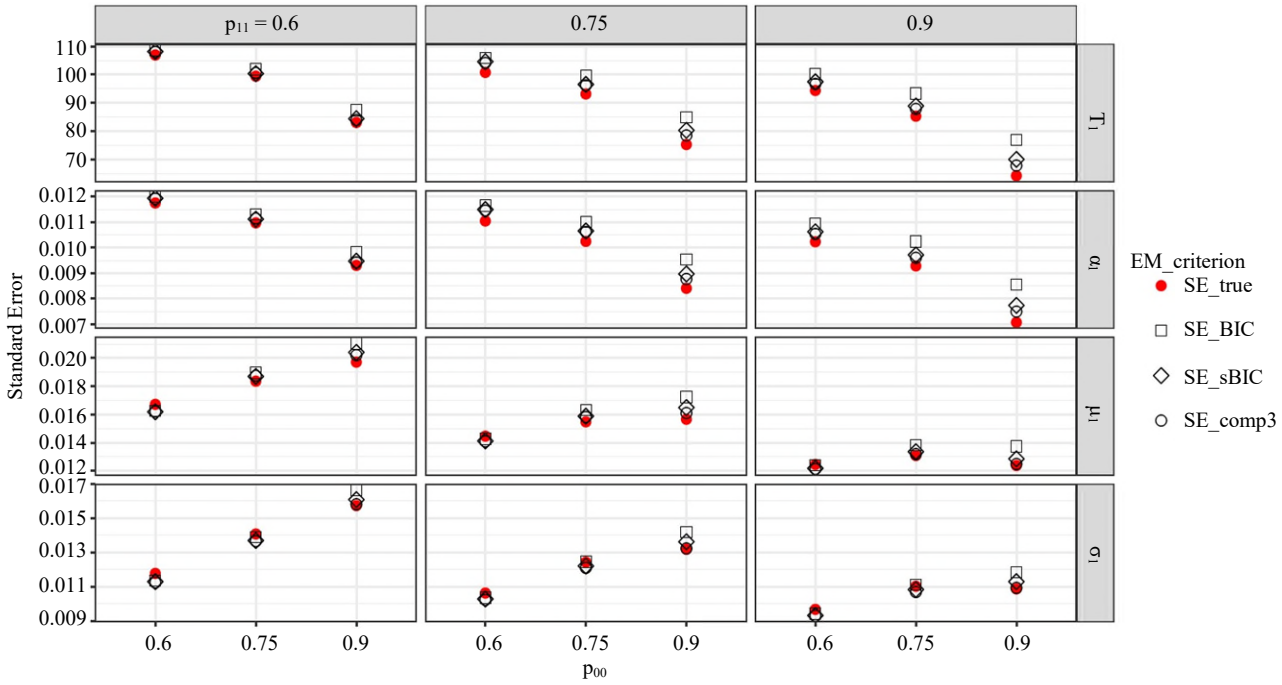
For case 2, we also tried a model with three components in both classes instead of two. The resulting bias estimates and standard errors are also shown in Figures B.1 and B.2 ("comp3"). As mentioned in Section 4.3, it is seen that increasing the number of components to three led to more accurate results.

Figure B.1 Bias estimation using BIC vs. sBIC in case 2.



Notes: For row and column settings see Figure A.1.

Figure B.2 Standard error estimation using BIC vs. sBIC in case 2.



Notes: For row and column settings see Figure A.1.

C. Theoretical properties of Algorithms 1 and 2

C.1 Algorithm 1: The bootstrap method

In general, Algorithm 1 yields biased estimates of the bias and variance of $\hat{\zeta}$. We will illustrate this using the estimated domain total $\hat{T}_1$ as an example. Note that the results below also apply to $\hat{\alpha}_1$, since it is obtained as a special case of $\hat{T}_1$ with $y_i \equiv 1/N$.

For $\hat{T}_1$, it can be derived that its true bias equals

$$\text{Bias}(\hat{T}_1) = (1-p_{00})\,T_0 + (p_{11}-1)\,T_1, \tag{C.1}$$

whereas, for $S \to \infty$, the bootstrap bias estimator from Algorithm 1 converges to $(1-p_{00})\,\hat{T}_0 + (p_{11}-1)\,\hat{T}_1$; see, e.g., Burger, van Delden and Scholtus (2015). Thus, in general, the bootstrap bias estimator is biased unless $\hat{T}_1$ itself happens to be an unbiased estimator of T_1. The latter situation occurs only for particular combinations of (p_{11}, p_{00}) (Kloos et al., 2021).

Similarly, it can be derived that the true variance of $\hat{T}_1$ is

$$\text{Var}(\hat{T}_1) = p_{00}(1-p_{00})\,K_0 + p_{11}(1-p_{11})\,K_1, \tag{C.2}$$

with $K_1 = \sum_{i=1}^{N} z_i y_i^2$ and $K_0 = \sum_{i=1}^{N} (1-z_i)\,y_i^2$, whereas the bootstrap variance estimator converges to the same expression with K_1 and K_0 replaced by $\hat{K}_1 = \sum_{i=1}^{N} \hat{z}_i y_i^2$ and $\hat{K}_0 = \sum_{i=1}^{N} (1-\hat{z}_i)\,y_i^2$, respectively

(Burger et al., 2015). Note that, in general when $T_1 \neq \alpha_1$, the conditions under which the bootstrap bias estimator and bootstrap variance estimator are unbiased are not equivalent.

C.2 Algorithm 2: The EM bootstrap method

As suggested in Section 2.3.2, the purpose of generating 0-1-values $\tilde{z}_i$ in the outer for-loop of Algorithm 2, with $P(\tilde{z}_i = 1 \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}}) = P(z_i = 1 \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}}) = \sum_{j=1}^{q_1} A_{1ji}$, is to correct the bootstrap bias and variance estimators for the bias that occurs in Algorithm 1. To illustrate the underlying idea, we will show here that in the case of the estimated domain total $\hat{T}_1$, for which exact analytical expressions are available, bias correction is indeed achieved by both $\mathbf{Bias}_{\mathbf{comb}}$ and $\mathbf{Var}_{\mathbf{comb}}$ provided that the assumed mixture model holds. A simplified version of the derivation below can be given for the estimated proportion $\hat{\alpha}_1$.

Denote $\tilde{T}_1 = \sum_{i=1}^{N} \tilde{z}_i y_i$ and $\tilde{T}_0 = \sum_{i=1}^{N} (1 - \tilde{z}_i)\, y_i$; also denote $\tilde{T}_1^* = \sum_{i=1}^{N} \tilde{z}_i^* y_i$ and $\tilde{T}_0^* = \sum_{i=1}^{N} (1 - \tilde{z}_i^*)\, y_i$. It can be derived analogously to (C.1) that $\mathrm{E}(\tilde{T}_1^* - \tilde{T}_1 \mid \hat{z}, \tilde{z}) = (1 - p_{00})\, \tilde{T}_0 + (p_{11} - 1)\, \tilde{T}_1$. Hence, for $S_1, S_2 \to \infty$, the expected value of the bias estimator in the EM bootstrap method is

$$\mathrm{E}\left\{\mathbf{Bias}_{\mathbf{comb}}(\hat{T}_1)\right\} = (1 - p_{00})\, \mathrm{E}\left\{\mathrm{E}_{\tilde{z}}(\tilde{T}_0 \mid \hat{z})\right\} + (p_{11} - 1)\, \mathrm{E}\left\{\mathrm{E}_{\tilde{z}}(\tilde{T}_1 \mid \hat{z})\right\}. \tag{C.3}$$

Furthermore, it is seen that

$$\mathrm{E}_{\tilde{z}}(\tilde{T}_1 \mid \hat{z}) = \sum_{i=1}^{N} \mathrm{E}(\tilde{z}_i \mid \hat{z}_i)\, y_i = \sum_{i=1}^{N} P(z_i = 1 \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}})\, y_i = \sum_{i=1}^{N} \sum_{j=1}^{q_1} A_{1ji} y_i = N \hat{\alpha}_1 \sum_{j=1}^{q_1} \hat{\xi}_{1j} \hat{\mu}_{1j},$$

where the last expression follows from the formulas applied during the M step of the EM algorithm (see Section 2.2.2). Under the assumption that the mixture model holds, it follows that

$$\mathrm{E}\left\{\mathrm{E}_{\tilde{z}}(\tilde{T}_1 \mid \hat{z})\right\} = \mathrm{E}\left(N \hat{\alpha}_1 \sum_{j=1}^{q_1} \hat{\xi}_{1j} \hat{\mu}_{1j} \right) \approx N \alpha_1 \sum_{j=1}^{q_1} \xi_{1j} \mu_{1j} = N \alpha_1 \mu_1 = T_1;$$

cf. note 2 at Table 2.1 for the final two equalities. Similarly, it can be shown that $\mathrm{E}\left\{\mathrm{E}_{\tilde{z}}(\tilde{T}_0 \mid \hat{z})\right\} \approx T_0$ if the model holds. Substituting both results into (C.3) and recalling (C.1), we conclude that $\mathrm{E}\{\mathbf{Bias}_{\mathbf{comb}}(\hat{T}_1)\} \approx \mathrm{Bias}(\hat{T}_1)$.

For the variance estimator, we can proceed in a similar fashion. Denote $\tilde{K}_1 = \sum_{i=1}^{N} \tilde{z}_i y_i^2$ and $\tilde{K}_0 = \sum_{i=1}^{N} (1 - \tilde{z}_i)\, y_i^2$. Analogously to (C.2) it can be shown that $\mathrm{Var}(\tilde{T}_1^* \mid \hat{z}, \tilde{z}) = p_{00}(1 - p_{00})\, \tilde{K}_0 + p_{11}(1 - p_{11})\, \tilde{K}_1$. Hence, for $S_1, S_2 \to \infty$,

$$\mathrm{E}\left\{\mathbf{Var}_{\mathbf{comb}}(\hat{T}_1)\right\} = p_{00}(1 - p_{00})\, \mathrm{E}\left\{\mathrm{E}_{\tilde{z}}(\tilde{K}_0 \mid \hat{z})\right\} + p_{11}(1 - p_{11})\, \mathrm{E}\left\{\mathrm{E}_{\tilde{z}}(\tilde{K}_1 \mid \hat{z})\right\}. \tag{C.4}$$

From the formulas applied during the EM algorithm, it follows that

$$\mathrm{E}_{\tilde{z}}(\tilde{K}_1 \mid \hat{z}) = \sum_{i=1}^{N} \mathrm{E}(\tilde{z}_i \mid \hat{z}_i)\, y_i^2 = \sum_{i=1}^{N} P(z_i = 1 \mid y_i, \hat{z}_i; \hat{\boldsymbol{\theta}})\, y_i^2 = \sum_{i=1}^{N} \sum_{j=1}^{q_1} A_{1ji} y_i^2 = N \hat{\alpha}_1 \sum_{j=1}^{q_1} \hat{\xi}_{1j} (\hat{\sigma}_{1j}^2 + \hat{\mu}_{1j}^2).$$

Hence, assuming that the mixture model holds, we obtain:

$$\mathrm{E}\left\{\mathrm{E}_{\tilde{z}}(\tilde{K}_1 \mid \hat{z})\right\} \;\approx\; N\alpha_1 \sum_{j=1}^{q_1} \xi_{1j}(\sigma_{1j}^2 + \mu_{1j}^2) \;=\; N\alpha_1(\sigma_1^2 + \mu_1^2) \;=\; K_1.$$

In the same way, it can be derived that $\mathrm{E}\{\mathrm{E}_{\tilde{z}}(\tilde{K}_0 \mid \hat{z})\} \approx K_0$. Thus, it is seen using (C.2) that $\mathrm{E}\{\mathbf{Var}_{\mathbf{comb}}(\hat{T}_1)\} \approx \mathrm{Var}(\hat{T}_1)$ if the mixture model holds.

References

Bakker, B.F.M., Van Rooijen, J. and Van Toor, L. (2014). The system of social statistical datasets of Statistics Netherlands: An integral approach to the production of register-based social statistics. *Statistical Journal of the IAOS*, 30, 411-424.

Bross, I. (1954). Misclassification in 2×2 tables. *Biometrics*, 10, 478-486.

Buonaccorsi, J.P. (2010). *Measurement Error: Models, Methods and Applications*. Chapman and Hall/CRC Press.

Burger, J., van Delden, A. and Scholtus, S. (2015). Sensitivity of mixed-source statistics to classification errors. *Journal of Official Statistics*, 31(3), 489-506.

Chang, J., and Hall, P. (2015). Double-bootstrap methods that use a single double-bootstrap simulation. *Biometrika*, 102, 203-214.

Cook, J.R., and Stefanski, L.A. (1994). Simulation-extrapolation estimation in parametric measurement error models. *Journal of the American Statistical Association*, 89(428), 1314-1328.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1-22.

Drton, M., and Plummer, M. (2017). A Bayesian information criterion for singular models. *Journal of the Royal Statistical Society*, 79(2), 323-380.

Edwards, J.K., Bakoyannis, G., Yiannoutsos, C.T., Mburu, M.W. and Cole, S.R. (2019). Non-parametric estimation of the cumulative incidence function under outcome misclassification using external validation data. *Statistics in Medicine*, 38(29), 5512-5527.

Edwards, J.K., Cole, S.R. and Fox, M.P. (2020). Flexibly accounting for exposure misclassification with external validation data. *American Journal of Epidemiology*, 189(8), 850-860. Retrieved from https://doi.org/10.1093/aje/kwaa011 doi: 10.1093/aje/kwaa011.

Efron, B., and Tibshirani, R.J. (1993). *An Introduction to the Bootstrap*. London: Chapman and Hall/CRC.

Eurostat (2008). *NACE Rev.2 Statistical Classification of Economic Activities in the European Community*. Retrieved from https://ec.europa.eu/eurostat/ramon/nomenclatures/index.cfm?TargetUrl=LST_NOM_DTL&StrNom=NACE_REV2.

Eurostat (2009). *ESS Handbook for Quality Reports* (Tech. Rep.). Office for Official Publications of the European Communities, Methodologies and Working papers. Retrieved from https://ec.europa.eu/eurostat/web/products-manuals-and-guidelines/-/ks-ra-08-016.

Eurostat (2022). *Table: In-Work at-Risk-of-Poverty Rate by Educational Attainment Level - EU-SILC survey*. Retrieved from https://ec.europa.eu/eurostat/databrowser/view/ilciw04/default/table?lang=en.

Gravel, C.A., and Platt, R.W. (2018). Weighted estimation for confounded binary outcomes subject to misclassification. *Statistics in Medicine*, 37(3), 425-436.

Hall, P., and Martin, M.A. (1988). On bootstrap resampling and iteration. *Biometrika*, 75, 661-671.

Hopkins, D.J., and King, G. (2010). A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1), 229-247.

Keogh, R.G., Shaw, P.A., Gustafson, P., Carroll, R.J., Deffner, V., Dodd, K.W., Küchenhoff, H., Tooze, J.A., Wallace, M.P., Kipnis, V. and Freedman, L.S. (2020). Stratos guidance document on measurement error and misclassification of variables in observational epidemiology: Part 1-Basic theory and simple methods of adjustments. *Statistics in Medicine*, 39(16), 2197-2231. doi: 10.1002/sim.8532.

Kloos, K., Meertens, Q., Scholtus, S. and Karch, J. (2021). Comparing correction methods to reduce misclassification bias. In *Proceedings of BNAIC/BeneLearn*, (Eds., L. Cao, W.A. Kosters and J. Lijffijt), 103-129. Springer, Leiden. Retrieved from https://bnaic.liacs.leidenuniv.nl/wordpress/wp-content/uploads/papers/BNAICBENELEARN_2020_Final_paper_64.pdf.

Kooiman, P., Willenborg, L. and Gouweleeuw, J. (1997). *PRAM: A Method for Disclosure Limitation of Microdata*. Research paper no. 9705. Voorburg/Heerlen: Statistics Netherlands.

Kosinski, A.S., and Flanders, W.D. (1999). Evaluating the exposure and disease relationship with adjustment for different types of exposure misclassification: A regression approach. *Statistics in Medicine*, 18(20), 2795-2808.

Küchenhoff, H., Mwalili, S.M. and Lesaffre, E. (2006). A general method for dealing with misclassification in regression: The misclassification SIMEX. *Biometrics*, 62(1), 85-96.

Li, Y. (2020a). *Bias Correction for Classification Errors*. Retrieved from https://www.researchgate.net/publication/362862838_Bias_Correction_for_Classification_Errors (Internship Report, Leiden University).

Li, Y. (2020b). *Estimating the Effect of Classification Errors on Domain Statistics* (Master's thesis, Leiden University). Retrieved from https://hdl.handle.net/1887/3280845.

Little, R.J., and Rubin, D.B. (2002). *Statistical Analysis with Missing Data* (2nd ed., Vol. 793). New York: John Wiley & Sons, Inc.

Magnusson, P., Palm, A., Branden, E. and Mörner, S. (2017). Misclassification of hypertrophic cardiomyopathy: Validation of diagnostic codes. *Clinical Epidemiology*, 9, 403.

McLachlan, G.J., and Peel, D. (2000). *Finite Mixture Models*. New York: John Wiley & Sons, Inc.

Meertens, Q., Diks, C., Van den Herik, H. and Takes, F. (2020). A data-driven supply-side approach for estimating cross-border internet purchases within the European Union. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 183(1), 61-90.

Oosterveen, V. (2020). *Notice the Noise: Detecting Misclassifications in Register Data* (Master's thesis, Utrecht University, the Netherlands). Retrieved from https://www.researchgate.net/publication/354533688_Notice_the_noise_detecting_misclassifications_in_register_data.

Rousseeuw, P.J., and Croux, C. (1993). Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424), 1273-1283.

Schwartz, J.E. (1985). The neglected problem of measurement error in categorical data. *Sociological Methods and Research*, 13(4), 435-466.

Selén, J. (1986). Adjusting for errors in classification and measurement in the analysis of partly and purely categorical data. *Journal of the American Statistical Association*, 81, 75-81.

Shaw, P.A., Gustafson, P., Carroll, R.J., Deffner, V., Dodd, K.W., Keogh, R.H., Kipnis, V., Tooze, J.A., Wallace, M.P., Küchenhoff, H. and Freedman, L.S. (2020). Stratos guidance document on measurement error and misclassification of variables in observational epidemiology: Part 2-More complex methods of adjustment and advanced topics. *Statistics in Medicine*, 39(16), 2232-2263. doi: 10.1002/sim.8531.

Sinclair, D.G., and Hooker, G. (2017). An Expectation Maximization algorithm for high-dimensional model selection for the Ising model with misclassified states. *arXiv preprint arXiv:1704.05995*. Retrieved from https://arxiv.org/abs/1704.05995.

United Nations (2015). *Guidelines on Statistical Business Registers*. New York and Geneva: United Nations.

Van Delden, A., Scholtus, S. and Burger, J. (2016). Accuracy of mixed-source statistics as affected by classification errors. *Journal of Official Statistics*, 32(3), 619-642. Retrieved from https://content.sciendo.com/view/journals/jos/32/3/article-p619.xml doi: https://doi.org/10.1515/jos-2016-0032.

Van Delden, A., Scholtus, S., Burger, J. and Meertens, Q.A. (2023). Accuracy of estimated ratios as affected by dynamic classification errors. *Journal of Survey Statistics and Methodology*, 11(4), 942-966. doi: https://doi.org/10.1093/jssam/smac015.

Van den Brakel, J., and Bethlehem, J. (2008). *Model-Based Estimation for Official Statistics* (Tech. Rep.). Retrieved from https://www.cbs.nl/nl-nl/achtergrond/2008/10/model-based-estimation-for-official-statistics.

Van den Hout, A., and Van der Heijden, P.G.M. (2002). Randomised response, statistical disclosure control and misclassification: A review. *International Statistical Review*, 70(2), 269-288.

Zhang, L.C. (2011). A unit-error theory for register-based household statistics. *Journal of Official Statistics*, 27(3), 415-432.