8-14-2019 11:30 AM

Classification with Measurement Error in Covariates Or Response, with Application to Prostate Cancer Imaging Study

Kexin Luo, *The University of Western Ontario*

Supervisor: He, Wenqing, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Statistics and Actuarial Sciences

Abstract

The research is motivated by the prostate cancer imaging study conducted at the University of Western Ontario to classify cancer status using multiple in-vivo images. The prostate cancer histological image and the in-vivo images were subject to misalignment in the co-registration procedure, which can be viewed as measurement error in covariates or response. We investigate methods to deal with this problem.

The first proposed method corrects the predicted class probability when the data has misclassified labels. The correction equation is derived from the relationship between the true response and the error-prone response. The probability for the observed class label is adjusted so it approaches the probability of the true label. A model can be built with the corrected class probability and the covariates for prediction purpose.

A weighted model method is proposed to construct classifiers with error-prone response. A weight is assigned to each data point according to its position, which indicates the data point's reliability. We propose the weighted models for different machine learning classifiers, such as logistic regression, SVM, KNN and classification tree. The weighted model incorporates the weight for each instance in the model building process, and the weighted classifiers trained with the error-prone data can be used for future prediction.

The misalignment in the co-registration procedure can also be treated as measurement error in covariates. A weighted data reconstruction method is proposed to deal with the corrupted covariates. The proposed method combines two moment reconstruction forms under different assumptions. We incorporate the weights of the data to build adjusted variables to replace the error-prone covariates. The classifiers can be trained on the reconstructed data set.

Numerical studies were carried out to assess the performance of each method, and the methods were applied to the prostate cancer imaging study. The results show all methods have significantly resolved the misalignment problem.

Keywords: Measurement error, misclassification, classification, imaging data, weighted model, machine learning, moment reconstruction.

Lay Summary

This research investigates three methods to improve the prostate cancer detection accuracy with medical images when the image data was not correctly measured.

The prostate cancer is the most common cancer among Canadian men, but the current detection methods suffer from low accuracy and high variability. Using medical images like MRI to build statistical models to predict cancer status is a promising solution. The prostate cancer image research team at the University of Western Ontario collected image data for this modelling purpose, but the data had measurement error. The error can be viewed as the cancer labels (response) are wrong or the image intensity measurements (covariates) are corrupted. Various previous studies have shown that these kinds of measurement errors decrease the prediction performance.

The first method we proposed builds the relationship between the true cancer status and the mislabelled status. Through this relationship we can correct the predicted cancer label.

We define the reliability of each data point by its position in the medical image. This reliability is a probability that reflects how likely this point is correctly measured. We propose to combine this reliability measure with the statistical models so that the new models are less vulnerable to the measurement error problem.

Last we propose to combine the reliability of the data with the moment reconstruction method proposed by Freedman et al. (2004). The moment reconstruction method creates an "adjusted" value for the error-corrupted covariate such that the "adjusted" value is close to the true value. The form of moment reconstruction depends on the assumption of the type of the error. We have found out that the prostate image measurement error corresponds to two different error types, and the reliability reflects how likely is each error type. We combined these two error types to create the adjusted values for the covariates, with the proportion for each error-type determined by the reliability.

The simulation studies and the real data application have shown the proposed methods significantly improve the prediction performance.

Acknowlegements

First of all, I would like to thank my supervisor Wenqing He. This thesis would not have been possible without the guidance and support of him. During the grad school, Dr. He has consistently been supportive, allowing me to grow as a statistician. Instead of telling me specifically what to do and how to do it, he allowed me to think by myself and develop my own research ability. Dr. He is a great advisor not only for his professional and scientific mentorship, but also because he gave me many suggestions on the future career.

I also owe a great degree of gratitude to Dr. Grace Yi. Her book has provided much useful information and guidance on my research. The regular data science meetings arranged by Dr. Yi and Dr. He were very interesting and insightful. These meetings remind me of how much I enjoyed the research in statistics. The parties held by Dr. Yi and Dr. He in Christmas or summer time were delightful memories.

I am fortunate to have many great friends in Western University, who not only supported my research, but also made my years here a lot of fun.

I want to thank the examination committee members Dr. Serge Provost, Dr. Hao Yu, Dr. Yayuan Zhu and Dr. Longhai Li for for their time, interest, and helpful comments.

Finally I want to thank my family for their love and encouragement. During the past five years, I have had many struggles and frustrations. It would be impossible for me to complete this thesis if it were not for their consistent support.

This work was supported in part by OICR Biostatistics Training Initiative (BTI) Studentship Award. I would like to thank Ontario Institute for Cancer Research for the two-year financial support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The prostate cancer imaging study

The prostate gland is a part of the male reproductive system. It adds nutrients and fluid to sperm. Prostate cancer is one of the most common cancers that affect Canadian men (Stewart et al., 2014). The known risk factors of prostate cancer are age, family history and diet, while obesity and some other factors like exposure to high levels of testosterone are possible risk factors (Stewart et al., 2014, chapter 5.11).

Prostate cancer can be slow-growing, and the signs and symptoms are not obvious in the early stage of the cancer (Filson et al., 2015). The common diagnose tests for prostate cancer include health history and physical exam, prostate-specific antigen (PSA) test, transrectal ultrasound (TRUS), biopsy, complete blood count, magnetic resonance imaging (MRI), bone scan, computed tomography (CT) scan (Alberts et al., 2015; Bonekamp et al.,2011; Makarov et al.,2012). Usually the patient suspected of prostate cancer takes prostate-specific antigen (PSA) test, which measures an enzyme in a man's blood produced exclusively by prostate cells (Alberts et al., 2015). A higher than normal PSA level can have many causes, and one of them is prostate cancer. To diagnose whether the patient with a high PSA level has prostate cancer, the patients may be asked to take medical images and/or biopsy.

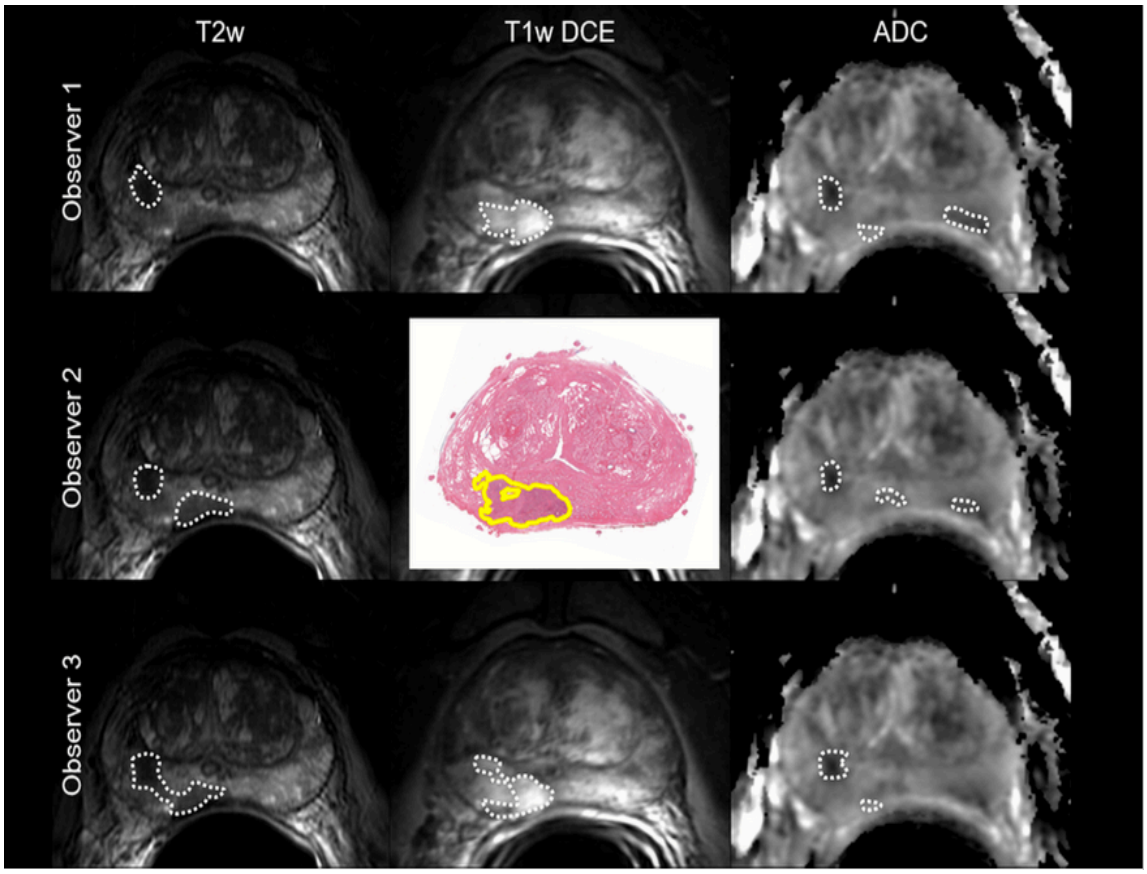


Figure 1.1: An illustration of the different diagnosis results with different observers and different image types.

The medical imaging techniques such as MRI and CT generate the images of the organs of the body, and the contours of possible cancer area marked by radiologists on these images help doctors to detect the cancerous lesions. However, the image diagnosis results vary largely upon both doctors' experiences and the types of medical images. Figure 1.1 shows the diagnosis results for different observers with different image types. The three horizontal rows represent three different observers, and the three vertical columns represent the three different imaging types (T2w, T1wDCE and ADC) for the same prostate slice. The white dashed circles in each image are the marked prostate cancer contour by certain observers with a certain medical image. The image in the centre is the histological image of that prostate slice, and the yellow contour is the exact cancer contour marked by pathologist and serves as the gold standard for cancer.

The marked cancer contours by different observers are different, and the same observer marked different contours on different medical images. None of the marked contour is the same as the true cancer contour on the histological image.

Biopsy involves extracting some sample cells or tissues from the body with a needle, and the cells or tissues are examined by pathologists to determine the presence or extent of a disease (Patel and Jones, 2009). The extraction of the cells or tissues in biopsy is usually guided with the 2D ultrasound, but depending on the experience of the technicians and the quality of the ultrasound image, the 2D ultrasound may not guide the needle to the correct location, thus the biopsy may not yield accurate results (see Pokorny et al., 2014, for instance).

To address the issues for the standard diagnosis procedure, there is a need to build a more reliable method to detect prostate cancer with limited access to the organ. One promising solution is to build correspondence between real cancer status and the in-vivo medical images of the prostate. Once the reliable relationship of prostate cancer status and medical images is established, it can be used not only to diagnose the prostate caner existence, but also to guide biopsy and targeted treatment for future patients.

In the prostate cancer imaging study conducted by the prostate cancer imaging research team at the University of Western Ontario, the researchers aimed to build statistical predict models to identify cancer position, size and grade with multiple medical images, such as MRI, CT and ultrasound image. This ability is crucial in the diagnosis and treatment to the patients.

The study protocol includes several steps. The in-vivo prostate images (multiple MRIs, ultrasound and CT) were taken for each patient, the prostate gland of each patient was then surgically removed from the body. After histoprocessing, ten strand-shaped fiducial markers were inserted into each specimen and the ex-vivo MR image of each prostate was taken. Then each prostate gland was sliced into three to five sections and the researchers took the histology image of each slice. The exact cancer contour on each slice was identified by pathologists using high resolution microscopes. The correspondence of the in-vivo images and the ex-vivo histological images was then conducted through a co-registration process (Gibson et al., 2012).

The co-registration process includes several steps. In the first step, the histology sections were reconstructed into a 3D ex-vivo context, by first using the fiducial markers to get an initial alignment between the histology and the ex-vivo MR images. The alignment was refined by a local optimization algorithm (Gibson et al., 2012) . In the second step, the ex-vivo MR image was registered to the in-vivo T2W MR image based on landmarks on the specimen (Ward et al., 2012). In the third step, the in-vivo T2W MR image was registered to other images such as the DCE and ADC images. After the registration, the multiple in-vivo medical images and the histology image were aligned for each position of the prostate. The whole prostate was then digitalized into voxels. Both cancer status and in-vivo image information were obtained for each prostate voxel with the aligned data.

(a) In-vivo MRI *(b) Histology image*

Figure 1.2: The in-vivo T2W MRI and histology image for the same prostate slice. (a) is the in-vivo 2DT2 image, (b) is the corresponding histology image. The coloured area in (b) is the diagnosed prostate cancer tissue.

As presented in Figure 1.2, the histology image (*b*), which clearly presents the exact cancerous tissue, serves as the response variable. The in-vivo image (*a*), which has measures including 2DT2, 3DT2, ADC and DCE of the prostate before it was taken out of the body, serves as the predictor variable. By building a model taking the in-vivo prostate image measurements as covariates and predicting the cancerous part on the histology image, the existence and position of prostate cancer for future patients can be predicted with the in-vivo image. The diagnosis and treatment of prostate cancer will then be expected to be largely improved and simplified.

However, the mapping of the histology image to the in-vivo image induced registration

error. In the mapping process of the ex-vivo image to the histology image (Gibson et al., 2012), ten fiducial markers were inserted into each prostate, and the mapping algorithm relied on finding the fiducial markers on the ex-vivo image and the histology image. This step depended largely on the experience of the researcher, and it was almost sure that there was a distance shifted in the mapping of the two images. Besides, the mapping algorithm assumed that an affine transformation exists between the histology and the ex-vivo images (Gibson et al., 2012). If this assumption was violated, which was usually the case, then the two images would not be perfectly co-registrated. The process of registering the ex-vivo image to in-vivo T2W MR image was landmark guided, which was more likely to induce registration error (Ward et al., 2010; Ward et al., 2012). The registration error was about 0.71 (0.38) mm from histology image to ex-vivo MR image, and 1.4 (0.2) mm from ex-vivo image to in-vivo T2W MR image. Since the T2W MR image was then registered to DCE and ADC, more measurement error was induced to ADC and DCE. The errors for this step were 1.0 (0.5) mm for DCE and 1.0 (0.2) mm for ADC. This registration error caused the true cancer status on the in-vivo images to shift for a certain distance from the registrated cancer status (Gibson et al., 2012).

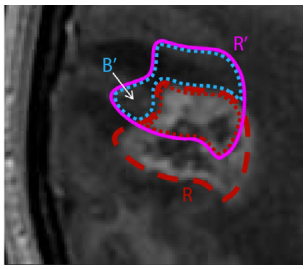


Figure 1.3: An illustration in Gibson et al. (2013). On the MRI of a brain, the true region of interest is R in red. Due to registration error, the sampling region is R′ (shown in purple). B′ (shown in cyan) is the background tissue.

Another aspect associated with this data set is that the cancer voxels only consist around 10% of all the data points. Measurement error combined with the highly imbalanced data may cause a much serious problem for classification.

The main objective of the prostate cancer imaging study is to build a reliable correspondence

between the cancer status and the in-vivo image information for each voxel of the prostate, meanwhile to account for the misalignment of the registration process. The first part corresponds to the classification problem, and the second part relates to the measurement error.

1.2 Review of classification methods

In this section some frequently used classification methods are briefly introduced, and the assessments of the classification performance are described.

For a voxel i, let Y_i denote a categorical response which can take K possible distinct values, such as cancer status (binary, $K = 2$) or cancer grade in the prostate cancer study (discrete, $K \geq 2$). Let $\boldsymbol{X}_i$ be a vector of covariates with dimension p representing the in-vivo image measurements. $\{\boldsymbol{x}_i, y_i\}$ is the realization of $\{\boldsymbol{X}_i, Y_i\}$, $i = 1, 2, \ldots, n$.

1.2.1 Logistic regression

The logistic regression models the $K - 1$ log-odds for the K classes with linear functions:

$$\begin{aligned}
\log \frac{\Pr(Y_i = 1|\boldsymbol{X}_i = \boldsymbol{x}_i)}{\Pr(Y_i = K|\boldsymbol{X}_i = \boldsymbol{x}_i)} &= \beta_{10} + \boldsymbol{\beta}_1^T \boldsymbol{x}_i \\
\log \frac{\Pr(Y_i = 2|\boldsymbol{X}_i = \boldsymbol{x}_i)}{\Pr(Y_i = K|\boldsymbol{X}_i = \boldsymbol{x}_i)} &= \beta_{20} + \boldsymbol{\beta}_2^T \boldsymbol{x}_i \\
&\vdots \\
\log \frac{\Pr(Y_i = K-1|\boldsymbol{X}_i = \boldsymbol{x}_i)}{\Pr(Y_i = K|\boldsymbol{X}_i = \boldsymbol{x}_i)} &= \beta_{(K-1)0} + \boldsymbol{\beta}_{K-1}^T \boldsymbol{x}_i,
\end{aligned}$$

where $\boldsymbol{\theta} = \{\beta_{10}, \boldsymbol{\beta}_1^T, \ldots, \beta_{(K-1)0}, \boldsymbol{\beta}_{K-1}^T\}$ is the vector of the regression coefficients. Denote $p_k(\boldsymbol{x}_i; \boldsymbol{\theta}) = \Pr(Y_i = k|\boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{\theta})$, then the log-likelihood for the data set is

$$L(\boldsymbol{\theta}) = \sum_i l_i(\boldsymbol{\theta}) = \sum_i \log p_{y_i}(\boldsymbol{x}_i; \boldsymbol{\theta}),$$

where $l_i(\boldsymbol{\theta}) = \log p_{y_i}(\boldsymbol{x}_i; \boldsymbol{\theta})$ is the log likelihood for each observation. Maximizing the log-likelihood gives the maximum likelihood estimation $\hat{\boldsymbol{\theta}} = \{\hat{\beta}_{10}, \hat{\boldsymbol{\beta}}_1^T, \ldots, \hat{\beta}_{(K-1)0}, \hat{\boldsymbol{\beta}}_{K-1}^T\}$ of $\boldsymbol{\theta}$.

To classify a new data point with $\boldsymbol{x}_{new}$, we calculate the K probabilities

$$\widehat{\Pr}(Y = k|\boldsymbol{X} = \boldsymbol{x}_{new}) = \frac{\exp(\hat{\beta}_{k0} + \hat{\boldsymbol{\beta}}_k^T \boldsymbol{x}_{new})}{1 + \sum_{l=1}^{K-1} \exp(\hat{\beta}_{l0} + \hat{\boldsymbol{\beta}}_l^T \boldsymbol{x}_{new})}, k = 1, \ldots, K-1,$$

$$\widehat{\Pr}(Y = K|\boldsymbol{X} = \boldsymbol{x}_{new}) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\hat{\beta}_{l0} + \hat{\boldsymbol{\beta}}_l^T \boldsymbol{x}_{new})},$$

and assign the data point to the class k for which $\widehat{\Pr}(Y = k|\boldsymbol{X} = \boldsymbol{x}_{new})$ has the largest value.

1.2.2 Support vector machine

Support vector machine (SVM) is a widely used classifier which produces decision boundaries to do classification (Boser et al., 1992; Cortes and Vapnik, 1995). The SVM classifier uses the kernel functions to map the original data set into a higher dimensional space, and finds a hyperplane which has the largest margin between the different classes in the projected space. This hyperplane serves as the future decision boundary. One attractive feature of SVM is that rather than using the whole data set, the hyperplane is determined by much fewer data points close to the hyperplane, termed as support vectors (Girosi, 1998).

In many situations, SVM classifier may not be able to separate the classes perfectly, but allow some misclassification in order to attain a larger margin. There is a trade-off between a larger margin and a smaller misclassification error, which can be controlled by a non-negative tuning parameter. To get a more flexible decision boundary with high dimensional features, the kernel function is used to extend the classifier to the enlarging feature space.

Specifically, consider the case that Y_i is binary, and takes values -1 and 1, the support vector classifier is a hyperplane defined by

$$\{\boldsymbol{x} : f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\beta} + \beta_0 = 0\},$$

where $\boldsymbol{\beta}$ is the vector of coefficients with unit value: $\|\boldsymbol{\beta}\| = 1$. The decision rule is

$$\text{sign}[\boldsymbol{x}^T \boldsymbol{\beta} + \beta_0].$$

The maximization of the margin is equivalent to the constrained optimization problem

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \sum_i \xi_i \right\}$$

subject to

$$\xi_i \geq 0, \quad y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i \quad i = 1, \ldots, n.$$

Here ξ_i is a non-negative value that controls the tolerance of observation i falling in the wrong side of the margin. The value of ξ_i larger than 1 leads to misclassification. The parameter C is the cost of violation, which controls the margin size.

The optimization problem can be rewritten into a Lagrangian dual objective function (Friedman et al., 2001):

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

subject to

$$\sum_i y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n$$

where α is the Lagrangian parameter, and

$$\boldsymbol{\beta} = \sum_i \alpha_i y_i \boldsymbol{x}_i.$$

The support vector machine extends the support vector classifier to the enlarged feature space by the kernel functions. The Lagrangian dual objective function can be expressed as

$$\begin{aligned} L_D &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle h(\boldsymbol{x}_i), h(\boldsymbol{x}_j) \rangle \\ &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \end{aligned}$$

subject to

$$\sum_i y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n$$

where C is the non-negative tuning parameter, $h()$ is the basis function to enlarge the feature space, $\langle \cdot, \cdot \rangle$ produces the inner product and $K()$ is the kernel function which can be expressed as

$$K(\boldsymbol{x}, \boldsymbol{x'}) = \langle h(\boldsymbol{x}), h(\boldsymbol{x'}) \rangle.$$

The decision function (separating hyperplane) can be written as

$$f(\boldsymbol{x}) = \sum_i \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + \beta_0,$$

where β_0 can be estimated by solving $y_i f(\boldsymbol{x}_i) = 1$ for any $\boldsymbol{x}_i$ with $0 < \alpha_i < C$. For a new observation $\boldsymbol{x}_{new}$, one calculates

$$\hat{f}(\boldsymbol{x}_{new}) = \sum_i \hat{\alpha}_i y_i K(\boldsymbol{x}_{new}, \boldsymbol{x}_i) + \hat{\beta}_0,$$

and assigns $\boldsymbol{x}_{new}$ according to the sign of $\hat{f}(\boldsymbol{x}_{new})$.

Some common choices of the kernel function $K()$ are

$$\begin{aligned} d\text{th} - \text{Degree polynomial} : K(\boldsymbol{x}, \boldsymbol{x}') &= (1 + \langle \boldsymbol{x}, \boldsymbol{x}' \rangle)^d, \\ \text{Radial basis} : K(\boldsymbol{x}, \boldsymbol{x}') &= \exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}'||^2), \\ \text{Neural network} : K(\boldsymbol{x}, \boldsymbol{x}') &= \tanh(\kappa_1 \langle \boldsymbol{x}, \boldsymbol{x}' \rangle + \kappa_2). \end{aligned}$$

The support vector machine can be extended into a regression method (Drucker et al., 1997). The basic idea of support vector regression is similar to support vector machine: maximize the margin of tolerance ϵ, which is the threshold that all fitted values must be within this range of the true values. Mathematically, this means solving

$$\max \begin{cases} -\frac{1}{2} \sum_i \sum_j (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ -\epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*) \end{cases}$$

subject to

$$\begin{cases} \sum_i (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

where y_i is a continuous number with covariate $\boldsymbol{x}_i$. The constant C is the tuning parameter. The SVM regression function can be written as

$$f(\boldsymbol{x}) = \sum_i (\alpha_i - \alpha_i^*) K(\boldsymbol{x}_i, \boldsymbol{x}) + \beta_0.$$

The intercept β_0 can be computed through KKT conditions (Kuhn, 1951; Karush, 1939).

1.2.3 Classification tree and random forest

The basic idea of a classification tree is to divide the feature space into some disjoint regions, and classify an observation based on the class of the region it belongs to (Breiman, 2001).

Specifically, suppose there are M regions $R_1, R_2, \ldots, R_M$, the classification tree has the form

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} c_m I(\boldsymbol{x} \in R_m),$$

where $I()$ is the indicator function, and c_m is the class label for region m. The value of c_m is determined by the dominant class for the points in region m. For example, if the region has more observations of cancer voxels (class 1), then the class label $\hat{c}_m$ for this region is 1 (cancer).

Consider the first split of the tree, and assume, for simplicity, the covariates $\boldsymbol{X} = (X_1, \ldots, X_p)$ are continuous. Two regions can be formed with covariate j and split point s:

$$R_1(j, s) = \{\boldsymbol{X} | X_j \leq s\} \quad \text{and} \quad R_2(j, s) = \{\boldsymbol{X} | X_j > s\}. \tag{1.1}$$

The two regions $R_1(j, s)$ and $R_2(j, s)$ are referred to as the nodes. The value of j and s can be solved by

$$\min_{j \in 1,\ldots,p,\ s \in P_j} (Q_1 + Q_2).$$

Here P_j can be any realized value of X_j, and Q_m, $m = 1, 2$ is the impurity measure for each of the two split nodes.

The popular choices of the impurity measure Q_m are

$$\begin{aligned}
&\text{Misclassification error:} && \frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)), \\
&\text{Gini index:} && \sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'}, \\
&\text{Cross-entropy or deviance:} && -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk},
\end{aligned}$$

where

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{\boldsymbol{x}_i \in R_m} I(y_i = k)$$

is the observed proportion of class k in node m with N_m observations and $k(m) = \arg\max_k \hat{p}_{mk}$ is the majority class in node m. After the first split, the same procedure is repeated to subsequent node until the terminal condition is met: the pre-set maximum number of nodes is achieved or the number of observations in a node is below the pre-set minimum number.

The tree method can be used to capture the complex structure of the data, and if it grows deep, the bias for the fitting can be very low. Yet the low bias would lead to high variation. Random forest classifier is proposed to improve the performance of classification tree by decreasing the variance of the prediction. It generates new training samples through bootstrap and builds a classification tree on each bootstrap sample. When constructing these classification trees, in each split only a small number of predictors are considered, which are randomly drawn from all the predictors. The prediction is the majority vote of the trees. This averaging idea of random forest can significantly reduce the variance of the classification tree, and in the same time preserve the low bias feature of the tree method.

The regression tree only changes the splitting criteria. The splitting variable j and splitting point s for the regions in (1.1) can be found by solving

$$\min_{j,s}\left\{\min_{c_1}\sum_{\boldsymbol{x}_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{\boldsymbol{x}_i\in R_2(j,s)}(y_i-c_2)^2\right\},$$

where

$$R_1(j,s)=\{\boldsymbol{X}|X_j\le s\}\ \text{ and }\ R_2(j,s)=\{\boldsymbol{X}|X_j>s\}$$

are the pair of half spaces of a splitting variable j and split point s, and y_i is a continuous response. Random forest regression is similar to the random forest classifier. The only difference is the output value for a random forest regression is the average value of the outputs for all the trees.

1.2.4 K-nearest neighbors

The K-nearest neighbors (KNN) classifier is a very simple yet powerful classification method. Given an observation, the KNN classifier first finds out the closest K data points to the observation, then the observation will be classified to the most common class among the K points. For continuous covariates, a commonly used distance is Euclidean distance. Hamming distance can be used for discrete covariates.

For a new observation $\boldsymbol{x}_{new}$, K nearest points are first identified, and the fitted value for this

new observation is obtained as

$$\hat{y} = \text{argmax}_r \left\{ \sum_{j=1}^{K} I(y_j = r) \right\},$$

where r is the set of all classes in the K data points, and $r = \{0, 1\}$ for binary class case.

KNN can be used for regression by changing the vote in the K points to the average value of their continuous responses.

1.2.5 Assessment of classification results

Some commonly used measures for evaluating the classification performance are introduced in this section.

Let $\hat{y}$ denote the predicted class label, and y is the corresponding true class label. Classification error rate is defined by

$$err = \frac{1}{n} \sum_{i}^{n} (\hat{y}_i \neq y_i),$$

where n is the number of responses predicted. Classification error rate measures the proportion of predicted results that are in the wrong class. It is an overall error and a straightforward measure that indicates how good the classifier performs for all the classes.

Sometimes the overall error may not be good enough, especially for imbalanced data. The sensitivity and specificity work as a pair that measures how well each class is correctly classified in a binary classification problem. Sensitivity is defined as

$$\frac{\text{number of true positives}}{\text{number of true positives + number of false negatives}},$$

and specificity is defined as

$$\frac{\text{number of true negatives}}{\text{number of true negatives + number of false positives}}.$$

The terms “true positive”, “true negative”, “false positive”, and “false negative” are generally used in medical data classification. In the prostate cancer prediction scenario, the relations of the terms can be viewed in Table 1.1.

	true cancer status		
predict		cancer	no cancer
cancer	cancer	true positive	false positive
status	no cancer	false negative	true negative

Table 1.1: The relationship of true positive, true negative, false positive and false negative in the prostate cancer prediction scenario.

In the prostate cancer scenario, sensitivity measures the proportion of true cancer voxels that are correctly classified, and specificity measures the proportion of non-cancer voxels that are correctly identified.

F1 score and G score measure the agreement between true response and predicted response. F1 score is calculated by

$$\text{F1} = 2\frac{\text{sensitivity} \cdot \text{precision}}{\text{sensitivity} + \text{precision}},$$

where precision is

$$\frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false positives}}.$$

The G score is defined by

$$\text{G} = \sqrt{\text{sensitivity} \cdot \text{precision}}.$$

F1 score is the harmonic mean of sensitivity and precision, while G score is the geometric mean.

Classification error rate serves as an overall measure of classification performance, a low value of classification error rate usually indicates a good classifier. Yet classification error rate does not tell how each class is correctly classified, and for imbalanced data set, classification error rate can be misleading. Sometimes the researchers may focus more on one class than the other. For example, in the prostate cancer data, more interest is put on correctly predicting the cancer voxels, other than the non-cancer voxels. In this case, one can refer to the sensitivity and

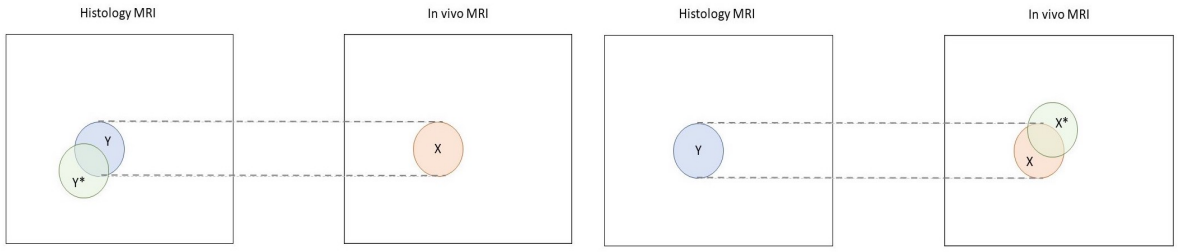
specificity. F1 score and G score also focus on the classification performance for each class. A high F1 score or G score (close to 1) indicates there is a high agreement between the predicted results and the true response. If the F1 score or G score get close to 0, there must be a huge difference between the predicted value and the true value. In this study, low error rate, high sensitivity, F1 score and G score will be of interest.

1.3 Review of measurement error models

In real application it is well known that epidemiology data is particularly common with the problem of measurement error (Michels, 2001), both in covariates and in response. In Yi (2017) the term "measurement error" is defined as any situation that the observed value of a variable is different from its true value. If the error-prone variable is a categorical variable, then one may use the term misclassification to refer to the measurement error problem.

In the prostate cancer imaging study, the misalignment issue can be viewed in two aspects. First, since the registered cancer status is shifted from the true cancer status on the in-vivo image, the observed cancer status on some voxels may be wrong. This can be viewed as an error in response, or specifically, misclassification in response. Another aspect to view this problem is that, for a voxel on the prostate, the true covariates corresponding to the voxel is shifted by a distance. As a result, if the response is treated error free, then the covariates are measured with error. Figure 1.4 shows the two ways of viewing the misalignment problem in the prostate cancer image data. In plot (a), the misalignment causes the true covariates in the in-vivo image mapping to Y^* instead of Y. In plot (b) the true cancer status Y in the histology data corresponds to the covariates X in the in-vivo data, while the misalignment leads to the shift of the covariates such that Y now is mapped to X^* instead.

From the definition of measurement error and misclassification it is easy to see the causes of measurement error vary greatly. Accordingly, the models of measurement error need to be flexible. In the following, we introduce some frequently used measurement error and



(a) *(b)*

Figure 1.4: Two ways of understanding the measurement error in the prostate cancer image data.

misclassification models.

1.3.1 Measurement error in covariates

Denote the true covariate $\boldsymbol{X}$, and the surrogate observation of $\boldsymbol{X}$ to be $\boldsymbol{X}^*$, then the classical additive error model can be expressed as

$$\boldsymbol{X}^* = \boldsymbol{X} + \boldsymbol{e},$$

where the error term $\boldsymbol{e}$ is assumed with mean $\mathbf{0}$ and covariance matrix $\Sigma_{\boldsymbol{e}}$, and is independent of $\boldsymbol{X}$. Since $Var(\boldsymbol{X}^*) = Var(\boldsymbol{X}) + \Sigma_{\boldsymbol{e}}$, the error-prone variable $\boldsymbol{X}^*$ in this model is more variable than the true covariate $\boldsymbol{X}$.

If the above model is written in an opposite form

$$\boldsymbol{X} = \boldsymbol{X}^* + \boldsymbol{e},$$

then it is referred to as Berson model. Similar to the classical additive error model, the error term $\boldsymbol{e}$ is assumed with mean $\mathbf{0}$ and covariance matrix $\Sigma_{\boldsymbol{e}}$, and is independent of $\boldsymbol{X}^*$. As a result, $Var(\boldsymbol{X}) = Var(\boldsymbol{X}^*) + \Sigma_{\boldsymbol{e}}$, so the true value $\boldsymbol{X}$ in Berson model is more variable than the error-prone variable $\boldsymbol{X}^*$. The variation relationship may be used to specify the appropriate measurement error model.

The latent variable model is a mixture of the classical additive error model and Berson

model, yet yields more flexibility:

$$\boldsymbol{X} = \boldsymbol{u} + \boldsymbol{e}_C \quad \text{and} \quad \boldsymbol{X}^* = \boldsymbol{u} + \boldsymbol{e}_B,$$

where the latent variable $\boldsymbol{u}$ has mean $\boldsymbol{\mu_u}$ and covariance matrix $\Sigma_{\boldsymbol{u}}$. The error terms $\boldsymbol{e}_B$ and $\boldsymbol{e}_C$ have mean $\mathbf{0}$ and covariance matrix Σ_B and Σ_C respectively.

A more general form of the classical additive error model proposed by Eckert et al. (1997) called transformed additive model is given by

$$g(\boldsymbol{X}^*) = g(\boldsymbol{X}) + \boldsymbol{e},$$

where $g(\cdot)$ is a monotone function, and $\boldsymbol{e}$ is independent of $\boldsymbol{X}$.

The measurement error models introduced above are all additive models, similarly, multiplicative models may also be used (Iturria et al., 1999). A simple multiplicative model is of the form

$$\boldsymbol{X}^* = \boldsymbol{X}\boldsymbol{e},$$

where $\boldsymbol{e}$ is independent of $\boldsymbol{X}$ and has mean $\mathbf{1}$.

In the case that the error-corrupted variable $\boldsymbol{X}^*$ depends on some other error-free covariates, a regression model may be used to describe their relationship. Suppose an error-free covariate $\boldsymbol{Z}$ is related to the error-corrupted variable $\boldsymbol{X}^*$, then the regression model of measurement error can be assumed the form

$$\boldsymbol{X}^* = \boldsymbol{\alpha_0} + \Gamma_{\boldsymbol{x}}\boldsymbol{X} + \Gamma_{\boldsymbol{z}}\boldsymbol{Z} + \boldsymbol{e},$$

where $\boldsymbol{e}$ has mean $\mathbf{0}$ and covariance matrix $\Sigma_{\boldsymbol{e}}$, and is independent of $\boldsymbol{X}$ and $\boldsymbol{Z}$. $\boldsymbol{\alpha_0}$ is a $p_x \times 1$ vector, $\Gamma_{\boldsymbol{x}}$ is a $p_x \times p_x$ matrix, $\Gamma_{\boldsymbol{z}}$ is a $p_x \times p_z$ matrix, p_x and p_z are dimensions of $\boldsymbol{X}$ and $\boldsymbol{Z}$, respectively.

The regression model of measurement error can be viewed as an extension of the classical additive error model. Switching the position of $\boldsymbol{X}$ and $\boldsymbol{X}^*$ in the above equation gives the regression version of Berson model.

The foregoing models consider the case of a continuous variable. When the error-prone variable is a categorical variable, i.e. it takes discrete values, one may consider modelling the misclassification process.

1.3.2 Misclassification in response

In the case of misclassification in response or so called label noise, let Y denote the binary response variable. There are two commonly used methods to model the misclassification process. One method is to model the conditional probability $\Pr(Y = y|Y^* = y^*, \boldsymbol{Z})$, and another one is through $\Pr(Y^* = y^*|Y = y, \boldsymbol{Z})$. The first conditional probability assumes that Y depends on Y^*, and vice versa. These two conditional probabilities are called (mis)classification probabilities. In the prostate cancer image data case, Y can be viewed as the unobserved exact cancer status on the in-vivo image, and Y^* is the observed version of Y.

1.3.3 Analysis methods for data with measurement error

There has been extensive research in the field of measurement error, and a large amount focuses on correcting the bias of the parameter estimation caused by measurement error (Yi, 2017 and Carroll et al.,2006). Yi (2017) discussed induced model method (observed likelihood method). The induced model method or observed likelihood method first models the relationship of the response Y and the observed error-prone covariate $\boldsymbol{X^*}$ and error-free covariate $\boldsymbol{Z}$ through the underlying response distribution model and the measurement error model. This relationship is used to construct the likelihood and the parameters is estimated by maximizing the likelihood. The EM algorithm incorporating measurement error is a simplification of the induced model method/observed likelihood method. When estimating the parameters, instead of maximizing the complete likelihood, EM algorithm simplifies the process by separating the optimization into E step and M step. Iteratively applying the E step and M step leads to the converged parameters. The conditional score method (Lindsay, 1982) further simplifies the estimation process by using a complete sufficient statistic instead of all the covariates in the E step.

The preceding methods can be viewed as likelihood-based correction methods, whose full distribution form for the response process is necessary. The unbiased estimating function methods relaxes the condition by requiring only the unbiased estimating functions in the estimation

process. For example, the subtraction correction method, discussed by Yi and Reid (2010), corrects the estimating function based on the error-prone covariate by subtracting the conditional expectation of the observed estimating function. A related method, expectation correction method was proposed to build a workable estimating function by calculating the expectation of the true estimating function conditional on the observed covariates and response. It has been proved that this estimating function is an unbiased estimate of the error-free estimating function. Another method called insertion correction method is opposite to the expectation correction method. The basic idea of insertion correction method is to find a computable estimating function for parameter estimation. As long as the conditional expectation of this estimating function, given the true covariate, error-free covariate and response, is the same as the estimating function of interest, consistent estimation of the parameters can be obtained.

A third class of correction methods is to directly correct the naive estimators which are obtained by treating the $\boldsymbol{X}^*$ as $\boldsymbol{X}$ and carrying out the usual estimating procedure. The naive estimator correction strategy, discussed by Stefanski and Carroll (1985), Yi and Reid (2010) and Yan and Yi (2016), is a typical way to correct the naive estimators. The basic idea of naive estimator correction strategy is to obtain a working estimator using estimating equations with the observed error data. Then the relationship between the true estimator and the working estimator is investigated, assuming the true data is known. The working estimator can then be corrected through the relationship established. Another approach called simulation-extrapolation (SIMEX), which was proposed by Cook and Stefanski (1994), can also be used to reduce the bias of the naive estimators. The SIMEX method first builds the trend of bias that induced by measurement error through simulation. Then the trend is extrapolated back to the case without measurement error.

Another class of correction method deals directly with the error-prone data. Prentice (1982), Carroll et al. (2006) discussed a method called regression calibration. This method replaces the error-prone covariate $\boldsymbol{X}^*$ with the conditional expectation $E(\boldsymbol{X}|\boldsymbol{X}^*, \boldsymbol{Z})$. Working with the new covariate $E(\boldsymbol{X}|\boldsymbol{X}^*, \boldsymbol{Z})$ instead of $\boldsymbol{X}^*$ reduces the bias. Based on the similar idea, Freedman et al.

(2004) introduced moment reconstruction method. This method is an extension of regression calibration that the error-prone variables are replaced by reconstructed values which retain the same first two moments of the error-free variables.

There are some other methods that correct the specific model. For example, Sexton and Laake (2007) proposed a method to estimate the true parameters in boosted regression trees with errors in covariates. The proposed method has a similar idea as insertion correction method, and it takes advantage of the specific form of the tree model to estimate the parameters.

The foregoing literature focuses on the estimation of model parameters for error in covariates. Yet much less attention has been paid to the impact of measurement error on prediction. Carroll et al. (2006) suggested that there is no need to worry about the impact of measurement error on prediction using linear models if future observations of predictors are also measured with error. When the covariate measurement error has a different distribution in the prediction set, Carroll et al. (2009) introduced a nonparametric method to estimate the prediction. Khudyakov et al. (2015) conducted numerical study to investigate the impact of covariate measurement error on risk prediction, which suggests that reducing measurement error in covariates improves the ensuing risk prediction.

There is an increasing discussion of misclassification in response or label noise in the recent years. Zhu and Wu (2004) pointed out that misclassification in response may cause worse results than measurement error in covariates. Yi (2017) discussed how misclassification on response may change the model structure. With univariate binary response with misclassification in a generalized linear model, Yi (2017) showed the link function changes from the error-free case. If the covariates are also measured with error, then even with certain simplified assumptions, the model of the observed data does not possess the same regression form as the true model. Neuhaus (1999) discussed how misclassification in response would induce bias and efficiency loss in coefficient estimation.

When the response has misclassification, there are roughly three approaches to deal with the problem. The first approach is to use a misclassification robust method for classification.

Although complete robustness is almost impossible to achieve (Frénay and Verleysen, 2013), there are some methods that work better than others with misclassification in response. In the paper of Folleco et al. (2008), the author compared 11 classifiers in the presence of misclassification in response, and random forest was shown to be the most robust method. Classification tree is a method that is sensitive to the misclassification in response (Abellán and Masegosa, 2010), but the imprecise info-gain as a node split criterion is shown to outperform other split criteria (Abellán and Masegosa, 2009) and makes the classification tree more robust.

The second approach to the misclassification in response problem is to filter and cleanse the training data. For example, Sun et al. (2007) proposed to use Bayesian classifier to estimate the probabilities for each instance falling in all possible classes, and the information entropy is then calculated. The instances with low entropy, but with a predicted label conflict to the observed label would be regarded as mislabeled cases. Miranda et al. (2009) proposed to train four different machine learning classifiers on the original data, then do a voting of the predictions of all these classifiers to detect the mislabeled cases. The voting filter can also be extended to local models. Sánchez et al. (2003) used the k-nearest centroid neighbors to predict the label of an instance while this instance is removed from the training set. If the predicted label is different from the observed label, then this instance is removed. Another method that use the property of AdaBoost was proposed by Verbaeten and Van Assche (2003). Since AdaBoost will increase the weight for the instances that cannot be well predicted, so the mislabeled instances will receive much larger weights in later iterations. Verbaeten and Van Assche (2003) proposed to remove the instances with highest weights after certain iterations.

The third approach is to combine the information of the mislabeled data in the modelling process. Eskin (2000) proposed a mixture model for the data: the instances are assumed generated either by a majority (normal) distribution or an anomalous distribution. At first all the instances are assumed in the majority class. Then for each instance, the change of the log likelihood of it is removed from the majority distribution and included in the anomalous distribution is calculated. If the difference of the log likelihood is deemed large enough, then

this instance is treated as an anomaly. Xu et al. (2006) proposed a robust SVM to deal with misclassification in response, which changes the loss function in the SVM objective to a more robust loss function. Yang et al. (2007) proposed a weighted SVM in the case of data set containing outliers and noises. The method calculates a weight between 0 and 1 for each data instance, measuring the reliability of the instance: a larger weight assigns to the more important instance. The weighted SVM then incorporates the weight in the slack variable ξ to control the level of violation of each point to the wrong side of the margin. Similarly, the weight idea can be used in the weighted KNN (Hechenbichler and Schliep, 2004), which finds the label for the new instance by the weighted votes of the nearest K points. If the distribution model of the data can be specified, then likelihood method, which models the likelihood of the observed data, can be used to find the true model parameters by maximizing the likelihood (Hausman et al., 1998). However, the integral calculating or approximating in likelihood method is usually a difficulty. To get around this problem, EM algorithms, or mean score method (Pepe et al., 1994) can be applied. Pepe (1992) extended the mean score method into more general settings, and Yi (2017) elaborated it into a semi-parametric method. Küchenhoff et al. (2006) extended the algorithm simulation and extrapolation (SIMEX) from measurement error in covariate to misclassification in response. Neuhaus (2002) presented how to deal with misclassification in response for clustered and longitudinal data. When misclassification arises in count response, Mwalili et al. (2008) proposed a method to correct for the the zero-inflated negative binomial regression model. When the response is continuous and contains measurement error, then least square method can be applied to find the true model (Yi, 2017; Sepanski and Lee,1995).

Yet the problem of the prostate cancer imaging study cannot be easily solved with the existing methods.

If the misalignment problem is viewed as the measurement error in covariates, the methods that deal with error in covariates are applied. The likelihood correction methods require the specification of the distribution form for the response process. The unbiased estimating function methods require the specification of the unbiased estimating function for the parameter

estimation. The distribution function or the estimating functions are very difficult or even impossible to specify for machine learning classifiers.

The methods to correct the naive estimators, such as SIMEX, assume that

$$\boldsymbol{X}_i^* = \boldsymbol{X}_i + \boldsymbol{e}_i, \quad i = 1, \ldots, n,$$

where $\boldsymbol{e}_i$ is independent of $\boldsymbol{X}_i$, follows a $N(0, \Sigma_{\boldsymbol{e}})$ distribution with the covariance matrix $\Sigma_{\boldsymbol{e}}$ known. This assumption is violated in the prostate cancer image data as the misalignment causes the true covariate observation to shift for a distance for each prostate voxel. If the covariates for a cancer voxel are shifted to a non-cancer covariates, then the error does not have mean 0. The regression calibration and moment reconstruction do not apply for the similar reason. These two methods assume that the error prone covariates $\boldsymbol{X}^*$ is an unbiased measurement of $\boldsymbol{X}$: $E(\boldsymbol{X}^*|Y) = E(\boldsymbol{X}|Y)$, which is not appropriate in the prostate cancer image data. The method proposed by Sexton and Laake (2007) to correct boosted regression tree with error in covariates is hard to apply since the method require the knowledge of $\Pr(\boldsymbol{X}|\boldsymbol{X}^*)$, and in the prostate cancer image data this relationship is difficult to model.

When the misalignment problem is treated as misclassification in response, the likelihood method, mean score method, and semi-parametric method are not applicable since the distribution forms or response models are hard or impossible to write out for machine learning classifiers. The data cleansing methods mentioned previously have the drawback of removing too many instances (Teng, 2000), and this problem becomes even more severe in the imbalanced dataset (Van Hulse and Khoshgoftaar, 2009). The extended SIMEX for misclassification in response (Küchenhoff et al., 2006) cannot be directly used here. The method assumes known and fixed misclassification probabilities for all points:

$$\pi_{ij} = \Pr(Y^* = i|Y = j),$$

However, when the responses are shifted for a distance, the misclassification probabilities for points in different position is different, so this assumption does not hold in the prostate cancer image data.

Another important feature of the prostate cancer imaging study is that the measurement error or misclassification in response relates to the spacial information of the data, since the error is caused by the misalignment of the images. Yet none of the above methods can take advantage of the spacial information of the image data.

1.4 Objectives and organization

In this research, the motivating prostate cancer imaging study has the misalignment problem in the registration process. The main interest of the study is to build a reliable model between the in-vivo medical images and histology cancer status for each voxel of the prostate so that in the future accurate detection and treatment of cancer tissues are possible with in-vivo image data.

Based on the above main interest, the goal of the thesis is to eliminate the influence of the misalignment for different classification methods. Both situations of misclassification in response and measurement error in covariates are considered and methods that work for different types of classifiers are of interest.

The rest of the thesis is organized as follows. In Chapter 2 we propose a predict probability correction method which corrects the predicted classification probability under the situation of misclassification in response. The classification probability of the observed class label given the covariate is corrected so that it is close to the probability of true class label given the covariates. Then by modelling the probability of true class label with the covariates, the classification probabilities for future observations can be calculated. A weighted model method is proposed in Chapter 3 which incorporates the weight for each data point under misclassification in response. The weight measures the reliability of each data point, and the weighted classifiers like weighted logistic regression, weighted KNN, weighted SVM and weighted classification tree are introduced. The weighted model construction process takes the reliability of each observation into consideration, and produces a classifier that can be used to classify future observations. In Chapter 4 we change the view to measurement error in covariates, and modify

the moment reconstruction method to combine it with the weight for each data point. Working with the reconstructed data eliminates the impact of measurement error on the model fitting. At last, the conclusion and future work is described in Chapter 5.

Chapter 2

Predict probability correction method

2.1 Introduction

In this chapter, we investigate the scenario that the response has misclassification and the covariates are error-free.

We propose a predict probability correction method that corrects the conditional probability of the observed class label given the predictors. The correction method constructs the relationship of the probabilities for the true response and the observed response using misclassification probabilities, so the classification probability for the error-prone response in the training set can be corrected through this relationship. The corrected probability is close to the conditional probability of the true class label given the predictors. A model can be built between the corrected probability and the covariates so that in the future this model can be used to predict the probability for each class given a new observation. The estimation methods for the misclassification probabilities in different scenarios are outlined. The numerical studies show the proposed correction strategy gives much better prediction results than those methods that ignore the misclassification in response.

The rest of the chapter is organized as follows. Section 2.2 describes the misclassification probabilities, and introduces the probability calculation for different classifiers. In section

2.3 the proposed predict probability correction method is presented and estimation of the misclassification probabilities in different settings is described. The simulation studies and the application to the prostate cancer imaging data are carried out to evaluate the proposed method for different classifiers in section 2.4. The chapter is concluded in section 2.5.

2.2 Notation and framework

Let $Y = \{0, 1\}$ denote the true binary response that may not be directly observed, and $\boldsymbol{Z}$ the vector of covariates with dimension p that is error-free. The observed version of Y is Y^*.

For instance i, there are two misclassification probabilities that are associated with Y_i and Y_i^* given covariate $\boldsymbol{Z_i}$, denoted by

$$\gamma_{10}(\boldsymbol{Z_i}) = \Pr(Y_i = 1|Y_i^* = 0, \boldsymbol{Z_i}) \quad \text{and} \quad \gamma_{01}(\boldsymbol{Z_i}) = \Pr(Y_i = 0|Y_i^* = 1, \boldsymbol{Z_i}). \tag{2.1}$$

In this chapter the classification probability is central to the proposed method. The calculation of the probability for predicted class for different classifiers is discussed below.

Logistic regression

The definition of logistic regression can be found in 1.2.1. In the binary scenario, the probability of getting class 1 is

$$\Pr(Y = 1|\boldsymbol{Z} = z) = \frac{\exp(\beta_0 + \boldsymbol{\beta}^T z)}{1 + \exp(\beta_0 + \boldsymbol{\beta}^T z)}.$$

SVM

As discussed in 1.2.2, the classification rule of support vector machine (suppose Y takes values -1 and 1) is the sign of the decision function

$$f(z) = \sum_i \alpha_i y_i K(z_i, z) + \beta_0. \tag{2.2}$$

In order to get a probability output for the SVM classifier, Platt (1999) proposed to fit a sigmoid function to the response and the value of f, and Lin et al. (2007) further improved it as:

$$\Pr(y = 1|f) = \frac{1}{1 + \exp(Af + B)},$$

where f is the fitted value of (2.2). The parameters A and B are estimated with the training set (f_i, y_i) by solving the negative log likelihood function:

$$\min_{A,B} \left\{ -\sum_i t_i \log(p_i) + (1 - t_i) \log(1 - p_i) \right\},$$

where

$$t_i = \frac{y_i + 1}{2} \quad \text{and} \quad p_i = \frac{1}{1 + \exp(Af_i + B)}.$$

Random forest

The random forest classifier is based on the classification tree (1.2.3). The classification tree divides the feature space into disjoint regions and a new observation is assigned with the class label according to the region it belongs to. To determine the classification result of a new observation using the random forest, the class labels of all trees in the forest are recorded, and the majority predicted class is assigned to the observation. The class probability is calculated as the proportion of that class in all trees (Malley et al., 2012).

KNN

The probability calculation for K-nearest neighbors (1.2.4) is similar to the averaging idea of random forest. When the nearest K points are found for the data point, the class label for that point is determined by the majority class in the K nearest points. The probability of the observation belongs to a class is the proportion of that class label in those K points (Malley et al., 2012).

2.3 Method description

2.3.1 Predict probability correction method

The relationship between the conditional probabilities Y_i^* given $\boldsymbol{Z}_i$ and Y_i given $\boldsymbol{Z}_i$ can be derived with the two misclassification probabilities $\gamma_{01}(\boldsymbol{Z}_i)$ and $\gamma_{10}(\boldsymbol{Z}_i)$:

$$\begin{aligned}
\Pr(Y_i = 1|\boldsymbol{Z}_i) &= \Pr(Y_i = 1, Y_i^* = 1|\boldsymbol{Z}_i) + \Pr(Y_i = 1, Y_i^* = 0|\boldsymbol{Z}_i) \\
&= \frac{\Pr(Y_i = 1, Y_i^* = 1, \boldsymbol{Z}_i)}{\Pr(Y_i^* = 1, \boldsymbol{Z}_i)} \frac{\Pr(Y_i^* = 1, \boldsymbol{Z}_i)}{\Pr(\boldsymbol{Z}_i)} + \frac{\Pr(Y_i = 1, Y_i^* = 0, \boldsymbol{Z}_i)}{\Pr(Y_i^* = 0, \boldsymbol{Z}_i)} \frac{\Pr(Y_i^* = 0, \boldsymbol{Z}_i)}{\Pr(\boldsymbol{Z}_i)} \\
&= \{1 - \Pr(Y_i = 0|Y_i^* = 1, \boldsymbol{Z}_i)\}\Pr(Y_i^* = 1|\boldsymbol{Z}_i) + \Pr(Y_i = 1|Y_i^* = 0, \boldsymbol{Z}_i)\Pr(Y_i^* = 0|\boldsymbol{Z}_i) \\
&= \{1 - \gamma_{01}(\boldsymbol{Z}_i)\}\Pr(Y_i^* = 1|\boldsymbol{Z}_i) + \gamma_{10}(\boldsymbol{Z}_i)\{1 - \Pr(Y_i^* = 1|\boldsymbol{Z}_i)\} \\
&= \gamma_{10}(\boldsymbol{Z}_i) + \{1 - \gamma_{01}(\boldsymbol{Z}_i) - \gamma_{10}(\boldsymbol{Z}_i)\}\Pr(Y_i^* = 1|\boldsymbol{Z}_i). \qquad (2.3)
\end{aligned}$$

The left hand side of (2.3) is the probability of interest. Once we know the probability $\Pr(Y_i|\boldsymbol{Z}_i)$, we can classify the data point to the class with a larger probability by the Bayes classification rule. The right hand side of (2.3) is workable as it only involves the known observed variables Y_i^* and $\boldsymbol{Z}_i$, and the equation does not require the knowledge of the specific model form. This equation provides a strategy to correct the predicted probability for the dataset with misclassification in response.

Given a training data set $\{(y_i^*, \boldsymbol{z}_i), i = 1, \ldots, n\}$, a classifier can be trained, and the probability $\Pr(Y_i^* = 1|\boldsymbol{Z}_i)$ can be obtained with the methods discussed in section 2.2. Based on equation (2.3), once the misclassification probabilities $\gamma_{01}(\boldsymbol{z}_i)$ and $\gamma_{10}(\boldsymbol{z}_i)$ are obtained, substituting $\Pr(Y_i^* = 1|\boldsymbol{Z}_i)$, $\gamma_{01}(\boldsymbol{z}_i)$ and $\gamma_{10}(\boldsymbol{z}_i)$ into the right hand side of (2.3) gives the estimate of $\Pr(Y_i = 1|\boldsymbol{Z}_i)$.

2.3.2 Estimation of misclassification probabilities

The model specification and estimation of $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ depend on the data generating process and assumptions. If the misclassification probabilities $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ depend on the covariates, a validation set is usually required to estimate them. The validation set contains $(y_i, y_i^*, \boldsymbol{z}_i)$, $i \in S_{vali}$, where S_{vali} is the index set of the data in the validation set.

$\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ usually can be modelled through a logistic regression to postulate their dependence on Z. For the validation set, a dummy variable I_{01i} is obtained as $I(y_i = 0|y_i^* = 1)$, $i \in S_{vali}$, where $I()$ is the indicator function. Fitting a logistic regression to $(I_{01i}, \boldsymbol{z}_i)$, $i \in S_{vali}$ gives the model of I_{01} and $\boldsymbol{Z}$: $\Pr(I_{01} = 1|\boldsymbol{Z}) = \exp(\beta_{001} + \boldsymbol{\beta}_{01}^T\boldsymbol{Z})/(1 + \beta_{001} + \boldsymbol{\beta}_{01}^T\boldsymbol{Z})$, and $\widehat{\Pr}(I_{01} = 1|\boldsymbol{Z})$ gives the estimate of $\gamma_{01}(\boldsymbol{Z})$. Similarly, one can replace the dummy variable I_{01i} to I_{10i} which is defined as $I(y_i = 1|y_i^* = 0)$, $i \in S_{vali}$, then the fitted logistic regression gives the model form of $\gamma_{10}(\boldsymbol{Z})$.

If more complex relationship of $\boldsymbol{Z}$ and misclassification probabilities is required, generalized additive model (GAM) or other non-linear models like random forest can be applied to capture the relationship through I_{01i} and I_{10i}.

In the prostate cancer image data, it is reasonable to assume that the misclassification probabilities γ_{01} and γ_{10} do not depend on the covariate $\boldsymbol{Z}$. In addition, although there is no true validation set for the prostate cancer image data available, the two dimensional coordinates that specify the position of each voxel are available. The misalignment of the in-vivo image and histology image can be viewed as the shift of the true cancer unit from the observed one. The mean and standard deviation of the shift distance were reported by Gibson et al. (2012), with the direction of the shift unknown. In order to estimate the misclassification probabilities γ_{01} and γ_{10} for the prostate cancer image data, we propose the following procedures.

For ith data point, a circle with the data point's coordinates being the centre and the shift distance being the radius is drawn. The weight ω_i is defined as

$$\omega_i = \frac{\sum_j \mathrm{I}(y_j^* = y_i^*)}{n_i}, \tag{2.4}$$

where j ranges among the voxels inside the circle, and n_i is the number of voxels in the circle of the ith observation.

Since the observed cancer status can be viewed as the true voxel shifted for a distance, the circle with the data point's coordinates as the centre and the shift distance as the radius indicates the area that the true response would be. If the circle contains more points with the same observed class label as for point i, then it is more likely that the true response y_i is the

same as the observed one y_i^*, and vice versa. This weight can be viewed as an estimate of the probability that measures how likely the true response Y is equal to the observed one Y^*. As a result, $1 - \omega_i$ can be viewed as the estimate of misclassification probability for point i. Specifically, if $y_i^* = 1$, then $\hat{\gamma}_{01} = \widehat{\Pr}(Y_i = 0|Y_i^* = 1) = 1 - \omega_i$, and $\hat{\gamma}_{10} = \widehat{\Pr}(Y_i = 1|Y_i^* = 0) = 0$; if $y_i^* = 0$, then $\hat{\gamma}_{10} = \widehat{\Pr}(Y_i = 1|Y_i^* = 0) = 1 - \omega_i$, and $\hat{\gamma}_{01} = \widehat{\Pr}(Y_i = 0|Y_i^* = 1) = 0$.

In the scenario of the prostate cancer imaging study, the point that is far away from the cancer and non-cancer boundary has a large weight. This point is more reliable than the one near the boundary since the misalignment is more likely to influence the points near the cancer and non-cancer boundary. Based on the definition, the points with weight 1 are surrounded by the same class within the shift region, so their class labels are not likely to be misclassified. By this method we are able to update the weight with the following steps.

- step 1: calculate the raw weights for all the points using equation (2.4);
- step 2: find the set M of points $(y_i^*, \boldsymbol{z}_i)$ with weight equal to 1 to build a preliminary classification model (such as logistic regression) between Y^* and $\boldsymbol{Z}$, denote the preliminary model by m_1;
- step 3: fit the preliminary model m_1 on the points with weight less than 1 (denoted set V), and get the estimated classification probability p^* that the fitted value $\hat{y}$ equals the observed one y^*;
- step 4: update the weights for the points in the set V. The new weight is set to be the probability p^* in step 3.

In step 2 the set M contains all points with weight 1, so it can be viewed as an error-free data set $(y_i, \boldsymbol{z}_i)$. Then the preliminary model m_1 is a preliminary classifier for $(Y, \boldsymbol{Z})$. Fitting this model on the covariates $\boldsymbol{z}_i$, $i \in V$ gives the preliminary predicted class label $\hat{y}_i$ for the points in the set V. If the probability that the predicted class label $\hat{y}_i$ being y_i^* is p_i^*, then we can treat this point with probability p_i^* to be correctly classified. The basic idea of the new weight

calculation method is using the correct data (the data with weight 1) to fit a rough model and then use the rough model to predict the reliability of the unsure data. The preliminary model, though not accurate enough, can still provide a better estimation to the unsure data than the raw weights because it contains more information.

In the case that there might not be enough data points with weight being 1 to build a preliminary model in step 2 due to the fact that the data set is very imbalanced, or the overlap between the true response Y and the observed response Y^* is very small, the preliminary model may not be very reliable. One possible solution is to use the points with weight being larger or equal to 0.9 or even 0.8. Another solution is to combine the new weights with the raw weights estimated by equation (2.4) by giving them different proportions, for example, the sum of the new weight and the raw weight each with weight 0.5 (but need to make sure the combined weight is in the range 0 to 1).

Denote the updated weight ω^*, then more accurate estimation of the misclassification probabilities can be found through ω^*: if $y_i^* = 1$, then $\hat{\gamma}_{01i} = \widehat{\Pr}(Y_i = 0|Y_i^* = 1) = 1 - \omega_i^*$, and $\hat{\gamma}_{10i} = \widehat{\Pr}(Y_i = 1|Y_i^* = 0) = 0$; if $y_i^* = 0$, then $\hat{\gamma}_{10i} = \widehat{\Pr}(Y_i = 1|Y_i^* = 0) = 1 - \omega_i^*$, and $\hat{\gamma}_{01i} = \widehat{\Pr}(Y_i = 0|Y_i^* = 1) = 0$.

2.3.3 Correction procedure

When only the in-vivo image is available to predict cancer status for future patients, the covariates $\boldsymbol{Z}$ can be viewed as error-free. As a result, we need to build a reliable model $\Pr(Y|\boldsymbol{Z}) = f(\boldsymbol{Z})$ with the available data set $(y_i^*, \boldsymbol{z}_i)$, $i = 1, \ldots, n$. The proposed correction procedure is as follows.

- step 1: fit the classifier on $(y_i^*, \boldsymbol{z}_i)$, $i = 1, \ldots, n$ and get the fitted probabilities $\widehat{\Pr}(Y_i^* = 1|\boldsymbol{Z}_i = \boldsymbol{z}_i)$;

- step 2: obtain the estimated values $\hat{\gamma}_{10i}$ and $\hat{\gamma}_{01i}$, $i = 1, \ldots, n$ by the methods described in 2.3.2;

- step 3: estimate the probability $\widehat{\Pr}(Y_i = 1|\boldsymbol{Z}_i = \boldsymbol{z}_i)$ with the value of $\widehat{\Pr}(Y_i^* = 1|\boldsymbol{Z}_i = \boldsymbol{z}_i)$, $\hat{\gamma}_{10i}$ and $\hat{\gamma}_{01i}$ by equation (2.3);

- step 4: a regression model can be built with $\widehat{\Pr}(Y_i = 1|\boldsymbol{Z}_i = \boldsymbol{z}_i)$ and $\boldsymbol{Z}_i$, $i = 1, \ldots, n$.

The model in step 4 is an estimate for $\Pr(Y = 1|\boldsymbol{Z})$, thus can be used in future prediction with covariate $\boldsymbol{Z} = \boldsymbol{z}_{new}$. The new observation can be classified to the class with a larger predicted probability.

In step 4 we need to capture the relationship between the corrected class probability and the covariate. Since the probability is a continuous variable ranges from 0 to 1, so regression models can be applied. The machine learning methods like SVM, random forest and KNN can be used to model the probability given the covariates, and the brief introduction of machine learning regression can be found in Chapter 1.2. Linear regression can also be employed to model this relationship. To ensure the predicted value lies between 0 and 1, we can model the log-odds of the corrected probability and the covariate with linear regression.

2.4 Numerical investigation

In this section we describe both the simulation studies and real data application of the proposed method. The numerical studies were done using R 3.5.2 (R Core Team, 2018). The packages *e1071* (Meyer et al., 2019), *randomForest* (Liaw et al., 2002), *FNN* (Beygelzimer et al., 2018), *class* (Venables and Ripley, 2002) and *kernlab* (Karatzoglou et al., 2004) were used to perform the corresponding analysis using SVM, random forest and KNN.

2.4.1 Simulation study

To evaluate the performance of the proposed predict probability correction method, simulation studies were carried out for a variety of scenarios.

Misclassification probabilities depend on covariates

In each run of the simulation study, a data set $(y_i, \boldsymbol{z}_i)$, $i = 1, \ldots, n$ with size $n = 1000$ or 5000 was simulated. Here Y was a binary response that took value 0 and 1. The covariate $\boldsymbol{Z} = (Z_1, Z_2)$ followed a bivariate normal distribution, and for each class of Y, the mean of $\boldsymbol{Z}$ was different while the variance was the same. Specifically, when $Y = 0$, $\boldsymbol{Z} \sim \text{Normal}(\boldsymbol{\mu}_0, \Sigma_{zz})$, and when $Y = 1$, $\boldsymbol{Z} \sim \text{Normal}(\boldsymbol{\mu}_1, \Sigma_{zz})$. In this simulation study, $\boldsymbol{\mu}_0$ was set to $(-0.8, 0.8)$, and $\boldsymbol{\mu}_1$ is $(0.8, -0.8)$. The variances of Z_1 and Z_2 were fixed at 1, and the correlation between the two covariates was 0.5. In this case, the two classes were separated by a 45 degree straight line. Different class proportions were considered in the simulation. Denote $\phi = \Pr(Y = 1)$ the proportion of class 1 in all the observations, and the value of ϕ was set to 0.2, 0.15 and 0.1.

In the first scenario, we considered that the misclassification probabilities γ_{01} and γ_{10} depend on the covariates $\boldsymbol{Z}$, i.e. $\gamma_{01} = \gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10} = \gamma_{10}(\boldsymbol{Z})$. In the simulation two methods were considered for the generation of $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$. First, the misclassification probabilities were generated through a linear form of $\boldsymbol{Z}$. The detailed generation process was as follows:

$$\gamma^*_{01}(\boldsymbol{Z}) = g^1_{01}(\boldsymbol{Z}) \quad \text{and} \quad \gamma^*_{10}(\boldsymbol{Z}) = g^1_{10}(\boldsymbol{Z}),$$

where in the situation that $\boldsymbol{Z}$ was a vector of covariates with order 2, i.e. $\boldsymbol{Z} = (Z_1, Z_2)$, we let

$$g^1_{01}(\boldsymbol{Z}) = Z_2 - Z_1 \quad \text{and} \quad g^1_{10}(\boldsymbol{Z}) = Z_1 - Z_2.$$

Linear transformations were employed to get $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ from $\gamma^*_{01}(\boldsymbol{Z})$ and $\gamma^*_{10}(\boldsymbol{Z})$:

$$\begin{aligned}
\gamma_{10}(\boldsymbol{z_i}) &= \lambda \frac{\gamma^*_{10}(\boldsymbol{z_i}) - \min\{\gamma^*_{10}(\boldsymbol{z_1}), \ldots, \gamma^*_{10}(\boldsymbol{z_n})\}}{\max\{\gamma^*_{10}(\boldsymbol{z_1}), \ldots, \gamma^*_{10}(\boldsymbol{z_n})\} - \min\{\gamma^*_{10}(\boldsymbol{z_1}), \ldots, \gamma^*_{10}(\boldsymbol{z_n})\}}, \\
\gamma_{01}(\boldsymbol{z_i}) &= \lambda \frac{\gamma^*_{01}(\boldsymbol{z_i}) - \min\{\gamma^*_{01}(\boldsymbol{z_1}), \ldots, \gamma^*_{01}(\boldsymbol{z_n})\}}{\max\{\gamma^*_{01}(\boldsymbol{z_1}), \ldots, \gamma^*_{01}(\boldsymbol{z_n})\} - \min\{\gamma^*_{01}(\boldsymbol{z_1}), \ldots, \gamma^*_{01}(\boldsymbol{z_n})\}},
\end{aligned} \tag{2.5}$$

where λ (< 1) was a parameter that controlled the overall error level $\epsilon = \Pr(Y \neq Y^*)$. In this simulation study λ was chosen so that the overall error level varied from 0.1 to 0.4. The equation (2.5) gives the misclassification probabilities $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$.

The nonlinear generation of misclassification probabilities was also considered, i.e., $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ did not depend on $\boldsymbol{Z}$ linearly. Let

$$\gamma^*_{01}(\boldsymbol{Z}) = g^2_{01}(\boldsymbol{Z}) \quad \text{and} \quad \gamma^*_{10}(\boldsymbol{Z}) = g^2_{10}(\boldsymbol{Z}),$$

where

$$g_{01}^2(\boldsymbol{Z}) = \begin{cases} \frac{1}{\sqrt{|\boldsymbol{Z_1}|+0.8}} + \frac{1}{\sqrt{|\boldsymbol{Z_2}|+0.8}} & \text{if } \boldsymbol{Z_2} > 0.9|\boldsymbol{Z_1}|, \\ 0 & \text{else,} \end{cases}$$

$$g_{10}^2(\boldsymbol{Z}) = \begin{cases} \frac{1}{\sqrt{|\boldsymbol{Z_1}|+0.8}} + \frac{1}{\sqrt{|\boldsymbol{Z_2}|+0.8}} & \text{if } \boldsymbol{Z_2} < -0.9|\boldsymbol{Z_1}|, \\ 0 & \text{else,} \end{cases}$$

for the situation where $\boldsymbol{Z}$ was a vector of covariates with order 2. The linear transformation (2.5) was applied to $\gamma_{01}^*(\boldsymbol{Z})$ and $\gamma_{10}^*(\boldsymbol{Z})$ to get the misclassification probabilities $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$.

The error-prone response Y^* was generated based on the misclassification probability $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$. After the misclassification probabilities $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ were set, Y^* was simulated according to the Bernoulli distribution: for individual i, a random number $q_i \sim U(0,1)$ was generated. If $Y = 1$, then $Y^* = 1 - I\{q_i < \gamma_{10}(\boldsymbol{Z})\}$, otherwise $Y^* = I\{q_i < \gamma_{01}(\boldsymbol{Z})\}$. The error level $\epsilon = \Pr(Y \neq Y^*)$ varied among 0.1, 0.2, 0.3 and 0.4.

The classifiers considered in this simulation study were K-nearest neighbours (KNN), support vector machine (SVM), random forest (RF) and logistic regression. The number of neighbours for K-nearest neighbours classifier was set to be 5. The number of trees to grow for random forest method was 500, and the number of parameters considered in each split was set to be the nearest integer of $\sqrt{p}$, where p was the dimension of the feature space. In the simulation setting p was equal to 2, so in each split only one covariate was considered. The kernel function for SVM was radial basis with gamma parameter being set to 0.5, and the cost being set to 100.

In step 4 of the correction procedure in 2.3.3, one needs to model the relationship of $\widehat{\Pr}(Y_i = 1|\boldsymbol{Z}_i)$ and $\boldsymbol{Z}_i$, $i = 1, \ldots, n$. In the simulation study, if the classifier used in step 1 was SVM, then SVM regression was used to model the corrected probability. If random forest classifier was used to do classification, then random forest regression was used to model the corrected probability. If KNN was the classifier used in step 1, then the corrected probability was modelled with KNN regression. If logistic regression was fitted to classify the training data in step 1, then we fitted the log-odds of the corrected probability and the covariate $\boldsymbol{Z}$ with

linear regression.

An independent sample of validation set $(y_i, y_i^*, \boldsymbol{z}_i)$ with size 100 or 200 was also generated with the same method of generating y_i and y_i^*. This validation set was used to estimate misclassification probability models $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ for the proposed method. Logistic regression or random forest were fitted on the validation set to estimate $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$.

In each simulation the generated data set was randomly separated into two folds with equal sizes, denoted S_{train} and S_{test}. S_{train} was used to train the classifier and S_{test} was used to test the performance of the method.

The following situations were simulated and tested in the study:

- T: the classifier is applied to the error-free training set $(y_i, \boldsymbol{z}_i)$, $i \in S_{train}$. The fitted model is tested on the testing set $(\boldsymbol{z}_i)$, $i \in S_{test}$.

- E: the classifier is applied to the error-corrupted training data set $(y_i^*, \boldsymbol{z}_i)$, $i \in S_{train}$. The fitted model is tested on the testing set $(\boldsymbol{z}_i)$, $i \in S_{test}$.

- C0: the proposed correction procedure described in 2.3.3 is applied to the error-corrupted training data set $(y_i^*, \boldsymbol{z}_i)$, $i \in S_{train}$, with the misclassification probabilities $\gamma_{01}(\boldsymbol{z}_i)$ and $\gamma_{10}(\boldsymbol{z}_i)$ being assumed known.

- C1: the proposed correction procedure described in 2.3.3 is applied to the error-corrupted training data set $(y_i^*, \boldsymbol{z}_i)$, $i \in S_{train}$, with the misclassification probabilities $\gamma_{01}(\boldsymbol{z}_i)$ and $\gamma_{10}(\boldsymbol{z}_i)$ being estimated with the validation set by logistic regression model.

- C2: the proposed correction procedure described in 2.3.3 is applied to the error-corrupted training data set $(y_i^*, \boldsymbol{z}_i)$, $i \in S_{train}$, with the misclassification probabilities $\gamma_{01}(\boldsymbol{z}_i)$ and $\gamma_{10}(\boldsymbol{z}_i)$ being estimated with the validation set by random forest model.

The measures of performance considered were classification error rate, sensitivity, specificity, F1 score and G score. Each scenario was repeated 1000 times, and the mean and standard deviation of the measures were recorded.

Simulation results for misclassification probabilities with linear form of Z

Table 2.1 reports the simulation results for KNN classifier with class 1 proportion being 15%, sample size being 5000, and the validation size being 200. Simulation results show that the classification performances depend largely on the error level. When misclassification probabilities depend linearly on the covariates, the error impact is not significant for error level ϵ being 0.1 or 0.2. The impact of misclassification in response is much more serious when the error level ϵ is larger than or equal to 0.3. The classification error rate has large increase, and sensitivity, F1 score and G score drop significantly. Different classifiers had similar performance for error-free data, but SVM and logistic regression were less vulnerable to the misclassification in response (except for logistic regression in the most imbalanced data scenario). Similar results were observed with class 1 proportion being 10% and 20%, but the imbalanced classes proportions made the impact of misclassification even larger.

The proposed correction procedure was employed to reduce the impact of misclassification in response. The simulation results show that the performance of the proposed method depends on the estimation accuracy of the misclassification error rate. When $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ are assumed known, the proposed correction method provides the classification performance close to the error-free scenario. When $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ are estimated through the validation set with logistic regression, the improvement is not significant for ϵ being 0.1 or 0.2. For ϵ equals 0.3 or 0.4, the proposed correction procedure provides much larger improvement over sensitivity, F1 score and G score, but the variation is also large. The proposed correction procedure had better performance on KNN and random forest when $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ depend linearly on $\boldsymbol{Z}$. The improvement for SVM and logistic regression was very limited except for the error level being 0.4, but this is not a big problem since SVM and logistic regression were not very vulnerable to the misclassification problem. Figure 2.1 shows the simulation results for KNN when $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ depend linearly on $\boldsymbol{Z}$, class 1 proportion being 0.15, and the sample size being 5000 with validation size being 200. The plots for other classifiers can be found in Figure 2.2, 2.3 and 2.4.

Decreasing the sample size from 5000 to 1000 did not change the conclusion, but the variation for all the measurements increased significantly (see Table 2.2).

Simulation results for misclassification probabilities with nonlinear dependence of Z

Table 2.3 shows the KNN classification results when the misclassification probabilities depend nonlinearly on the covariate $\boldsymbol{Z}$. The results indicate that the error impact is larger when misclassification probabilities depend nonlinearly on the covariate, compared with the linear scenario. When ϵ is 0.4, the F1 score would be less than one third compared to the error-free case when the misclassification error is ignored. The imbalanced class proportions also make the performance of all measures worse for all error levels.

Similar to the linear misclassification error scenario, the performance of the proposed correction procedure depended on the estimation of $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$. Figure 2.6 shows the simulation results for KNN when $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ depend nonlinearly on $\boldsymbol{Z}$, class 1 proportion being 0.15, and the sample size being 5000 with validation size being 200. When $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ are assumed known, there is a significant improvement with the proposed correction. In the scenario the misclassification probabilities are estimated with logistic regression in the validation set, some improvement is shown for sensitivity, F1 score and G score when ϵ is 0.3 or 0.4, but with quite large standard deviations. If the misclassification probabilities are estimated with random forest regression in the validation set, there is much more improvement for all error levels, and the standard deviations for the measures are smaller. This difference in performance is expected since the true $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ depend nonlinearly on $\boldsymbol{Z}$, so random forest is better to capture the relationship than logistic regression.

The validation size is crucial in the performance of the proposed correction method since the estimation of $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$ is the key. Larger validation set provides much larger and more stable improvement than a small validation set.

The simulation results for other classifiers are shown in Figure 2.5, 2.6, 2.7 and 2.8, the findings are similar to our report above.

Misclassification probabilities from misalignment

The data generation process was as follows.

The Cartesian coordinates (w_{1i}, w_{2i}), $i = 1, \ldots, n$ were generated with data size n being 1000 or 5000. w_1 and w_2 were simulated from a bivariate normal distribution, with mean being set to (0,0), and covariance an identity matrix. Denoting the cancer status as class 1, and the non-cancer status as class 0, the binary response Y was generated as $y_i = I(w_{1i}^2 + w_{2i}^2 \leq r^2)$, $i = 1, \ldots, n$. The value of r determined the class proportion $\phi = \Pr(Y = 1)$, and the value of ϕ ranged among 0.05, 0.1, 0.15, and 0.2. The layout of the simulated (w_1, w_2) mimics the shape of the prostate.

The covariate $\boldsymbol{Z} = (Z_1, Z_2)$ was simulated following a bivariate normal distribution with fixed variance and different means for each response class. If $Y = 0$, $\boldsymbol{Z} \sim \text{Normal}(\boldsymbol{\mu}_0, \Sigma_{zz})$, otherwise, $\boldsymbol{Z} \sim \text{Normal}(\boldsymbol{\mu}_1, \Sigma_{zz})$. The value $\boldsymbol{\mu}_0$ was set to $(-0.8, 0.8)$, and $\boldsymbol{\mu}_1$ was $(0.8, -0.8)$. The variances of Z_1 and Z_2 were 1, and the correlation between the two covariates was 0.5.

The misclassified response Y^* was simulated similar to the misalignment mechanism. For simplicity, only the cancerous lesion ($y = 1$) was shifted. The error-prone response Y^* was generated as $y_i^* = I\{(w_{1i} + a)^2 + (w_{2i} + a)^2 \leq r^2\}$, $i = 1, \ldots, n$, where a was a positive value that controlled the shift distance $\sqrt{2}a$. The value of a was determined by the overlap proportion of the true cancerous tissue ($y = 1$) and the observed value ($y^* = 1$). A small value of a shifts the observed response for a small distance, which leads to a large overlap between the area of $y = 1$ and $y^* = 1$. The overlap proportion was defined as $\Pr(Y^* = Y | Y = 1)$, and its values were set to 0.5, 0.6, 0.7 and 0.8.

In each run the data set was randomly divided into half training S_{train} and half testing S_{test}. The following scenarios were investigated:

- T: the classifier is fitted on the error-free training set $(y_i, \boldsymbol{z}_i)$, $i \in S_{train}$ and the fitted model is tested on the testing set $(\boldsymbol{z}_i)$, $i \in S_{test}$.

- E: the classifier is fitted on the error-corrupted training set $(y_i^*, \boldsymbol{z}_i)$, $i \in S_{train}$ and the fitted

model is tested on the testing set (z_i), $i \in S_{test}$.

- C1: the proposed correction procedure procedure described in 2.3.3 is applied to the training set (y_i^*, z_i), $i \in S_{train}$, with γ_{01} and γ_{10} being estimated by raw weight. The fitted model is tested on the testing set (z_i), $i \in S_{test}$.

- C2: the proposed correction procedure described in 2.3.3 is applied to the training set (y_i^*, z_i), $i \in S_{train}$, with γ_{01} and γ_{10} being estimated by updated weight. The fitted model is tested on the testing set (z_i), $i \in S_{test}$.

The predicted class label was compared to the true class label for the testing set. The measures of performance considered were classification error rate, sensitivity, specificity, F1 score and G score. Each scenario was repeated 1000 times, and the mean and standard deviation of the measures were recorded.

In the simulation study, logistic regression, SVM, KNN and random forest were considered. For KNN classifier, 5 nearest neighbors was considered. The number of trees to grow for random forest classifier was 500, and in each split one covariate was considered. The radius kernel function was used for SVM with gamma being set to 0.5, and the cost being set to 100. Similar to the scenario that the misclassification probabilities depend on the covariates, we applied SVM regression, random forest regression, KNN regression or linearly modelled the log-odds of the corrected probability in step 4 in the proposed correction procedure in 2.3.3.

Table 2.4 presents the simulation results for random forest classifier with data size being 5000. The different scenarios are summarized in the table. The overlap proportion has a great impact on the performance of the random forest classifier. Smaller overlap proportion produces worse classification results. The impact is also larger for more imbalanced data set. For all class 1 proportions, a low overlap proportion (50% or 60%) doubles or even triples the classification error rate, and the sensitivity drops to less than half compared to the error-free scenario. The F1 score and G score also decrease significantly. The impact of misalignment is much less for moderate and high overlap proportions (70% or 80%). For a relatively large class 1 proportion,

the 80% overlap produces only slightly worse classification results compared to the error-free scenario.

The classification results for different classifiers are shown in Figure 2.9, 2.10, 2.11 and 2.12. Different classifiers reacts differently to the misalignment problem. SVM and logistic regression are more vulnerable to low overlap proportions, especially when the overlap is only 50%.

The simulation results indicate that the estimation of the weight has a huge impact on the proposed correction procedure. If the weight is estimated by equation (2.4), no improvement or even worse results are observed with the proposed correction procedure (C1). Yet if the weight is updated, the proposed correction (C2) produces really good improvement. With updated weights, the classification error rate drops significantly compared to the scenario when the misalignment problem is ignored. The sensitivity, F1 score and G score all increase significantly. The only exception was SVM when the overlap proportion is relatively high (70% or 80%). In this case the proposed correction procedure does not provide improvement (see Figure 2.10).

The proposed correction procedure with updated weights produced the most significant improvement for random forest classifier. The improvement for KNN and logistic regression was also large. If the overlap proportion was only 50%, the standard deviations of sensitivity and F1 score for the proposed method were large, especially for logistic regression and random forest classifier. Table 2.5 indicates the sample size of 5000 or 1000 does not lead to different conclusions, but a larger sample size decreases the standard deviations for all measurements.

One drawback of the proposed method is that the model is not very robust when the true data is error-free but fitted under the correction procedure (see Table 2.6). The error induced by fitting the error-free data with the predict probability correction method was relatively large for small overlap proportion (50% or 60%), but negligible for large overlap proportion (70% or 80%).

2.4.2 Application to the prostate cancer image data

The proposed predict probability correction method was applied to the prostate cancer imaging study. The ongoing study was conducted by the research team supported by a team grand from the Canadian Institutes of Health Research. There were 43 patients who had been diagnosed prostate cancer enrolled in the study. The prostate gland for each patient was sliced into 3 to 5 slices, and each slice of the prostate had a histology MR image and several co-registered in-vivo MR images. The in-vivo MR images, which serve as the predictors, have intensity measures as 2DT2W, 3DT2W, ADC and DCE. These measures were standardized before analysis. The registration error induced in the mapping process of the histology image and the in-vivo image was measured as the 3D misalignment distance of the small anatomical landmarks identified on the histology and MR images. The mean registration error was 1.86 mm with standard deviation of 0.47 mm, according to the report from the research group (Gibson et al., 2012).

Since there is no testing set for the prostate cancer image study, the validation of the proposed method is difficult. We propose to construct the testing set by the following method. For each patient, one slice of the prostate images was isolated for testing, and the rest slices were used for training. On the isolated testing slice, the weight for each point was calculated given the shift distance (registration error) with the method introduced in 2.3.2. The points with weight 1 were by definition the points with correct cancer labels, so these points were grouped as the testing set.

Among the 43 patients, 32 had valid data sets for classification. In this 32 data sets, we applied the proposed method on 5 of them that have relatively large training and testing sets. In the real data fitting procedure, three different registration errors were considered: the mean registration error 1.86 mm, and the values at one standard deviation, i.e. 1.39 mm and 2.33 mm. The machine learning classifiers logistic regression, KNN, SVM and random forest were fitted and tested. In the application, the updated weights were used to estimate γ_{10} and γ_{01}.

The classification results are summarized in Table 2.7, 2.8, 2.9, 2.10 and 2.11. The proposed predict probability correction method shows improvement for almost all classifiers on at least

one registration error distance. With the proposed correction method, the classification error rate decreases, sensitivity, specificity, F1 score and G score increase. Specifically, the classification error rate drops for all classifiers applied on all patients, and the specificity increases for almost all classifiers and all patients.

For patient 1015, the data in the second slice of the prostate was used for testing (8761 to 8792 instances, depending on the registration error), and the data in the rest slices were used for training (23023 instances). Table 2.7 shows that almost all measures for all classifiers get improved with the proposed method. The improvement under three registration error assumptions does not differ much.

For patient 2008, the data in the third slice was used for testing (9709 to 9893 instances, depending on the registration error), and the rest slices were used for training (46950 instances). In Table 2.8, the classification results for patient 2008 show that the sensitivity and F1 score are largely improved under the proposed method, and the improvement varies under different registration error assumptions. For example, the assumption of registration error being 2.33 mm works best for patient 2008. Almost all classifiers have the most significant improvement under this assumption. As a result, it is very likely that for patient 2008, the registration error was close to 2.33 mm. In this scenario, the sensitivity under random forest classifier increases by over 38%, and the F1 score is almost doubled. The logistic regression performs unsatisfactorily in the classification, probably due to the non-linear relationship in the covariates and the cancer status. In this case, the proposed method was not able to provide any improvement since the probability estimation under logistic regression was not reliable.

The third slice of patient 1012 was used for testing (12257 to 12374 instances, depending on the registration error), and only the forth slice was used for training (11003 instances) since the first and second slices had very few cancer instances. In Table 2.9, the classification results get the most improvement with the proposed method under the assumption of 1.39 mm registration error, except for logistic regression. The increase of sensitivity is not large, but the error rate, specificity and F1 score are much improved.

The first and forth slice of patient 1035 were used for training (16639 instances), and the second slice was used for testing (8209 to 8569 instances, depending on the registration error). It can be seen from Table 2.10 that the proposed method improves almost all measures of all classifiers in all assumed registration error (except for logistic regression, which gets improved only with 2.33 mm registration error). The improvement is very large, especially for random forest classifier. For example, under the 1.86 mm registration error, the sensitivity of random forest raises from 0.643 to 0.990 with the proposed method, and the F1 score is more than doubled.

For patient 2009, the third slice was used for testing (14862 to 15268 instances, depending on the registration error), and the second and forth slice were used for training (33118 instances). Table 2.11 presents the classification results for patient 2009. It can be seen that the proposed method improves the classification results for KNN and random forest for all assumed registration distances, but logistic regression and SVM are barely improved.

The results of all the 5 patients indicated that random forest benefited the most from the proposed method, which was consistent with the simulation results. Logistic regression, by contrast, got the least improvement, partly due to the fact that the linear relationship did not hold for some patients.

2.5 Conclusion

In this chapter we come up with a predict probability correction method that can directly correct the predicted class probability for each data point in the case of misclassification in response. This correction method is built with the relationship of conditional probabilities $\Pr(Y^* = 1|\boldsymbol{Z})$ and $\Pr(Y = 1|\boldsymbol{Z})$ through the misclassification probabilities $\gamma_{01}(\boldsymbol{Z})$ and $\gamma_{10}(\boldsymbol{Z})$.

We propose a method to estimate γ_{01} and γ_{10} in the setting of misalignment of the images. The proposed correction procedure corrects the predicted probability $\Pr(Y^* = 1|\boldsymbol{Z})$ in the training set, and models the relationship of the corrected probability and the covariates. Sim-

ulation studies and real data application show that the proposed correction procedure provides improvement compared to training directly with $(Y^*, \boldsymbol{Z})$.

This predict probability correction method has the advantage of a very simple correction procedure. However, it is not very robust when the data set is error-free but corrected with the proposed method. Thus the existence of the misclassification error needs to be verified before the application of the proposed method.

2.6 Appendix

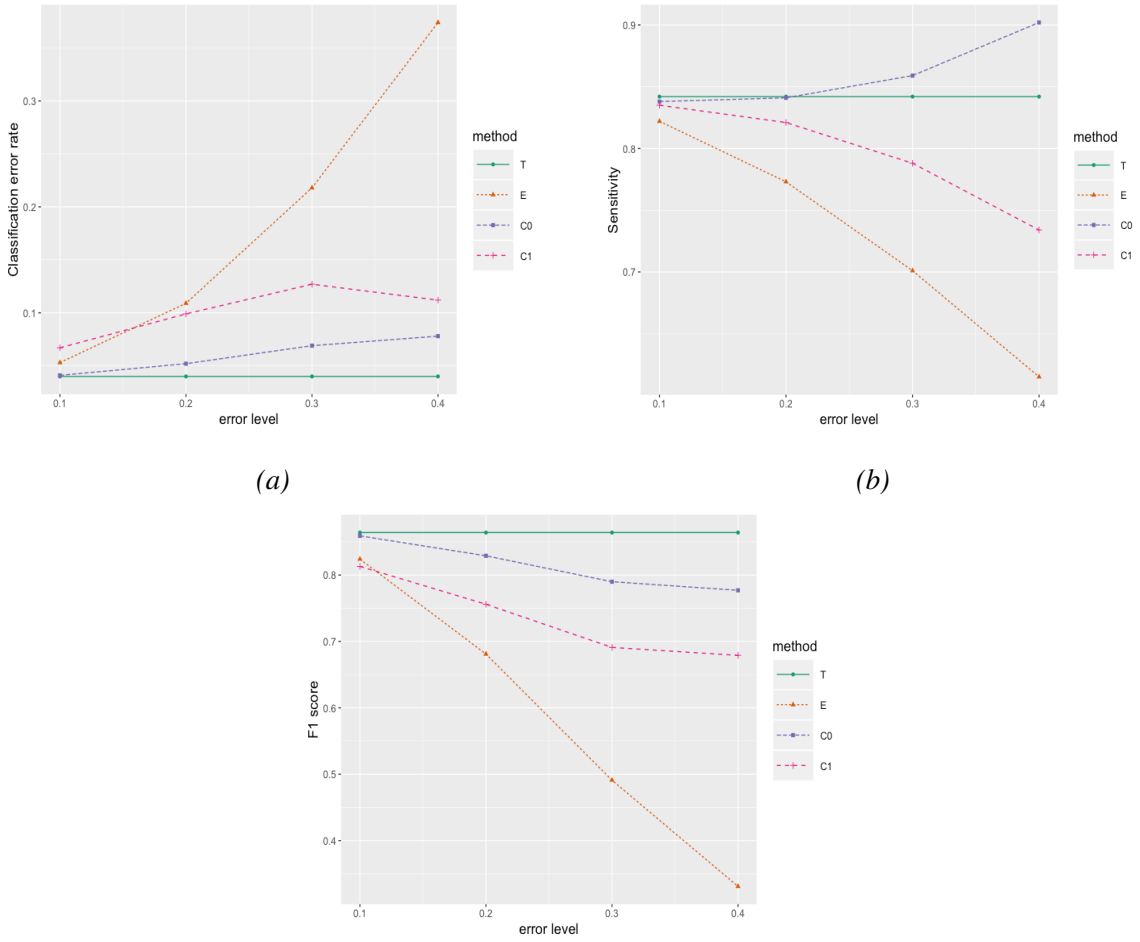


Figure 2.1: Simulation results for KNN with class 1 proportion ϕ=15%, sample size being 5000 and validation size being 200. The misclassification error depends linearly on the covariates.

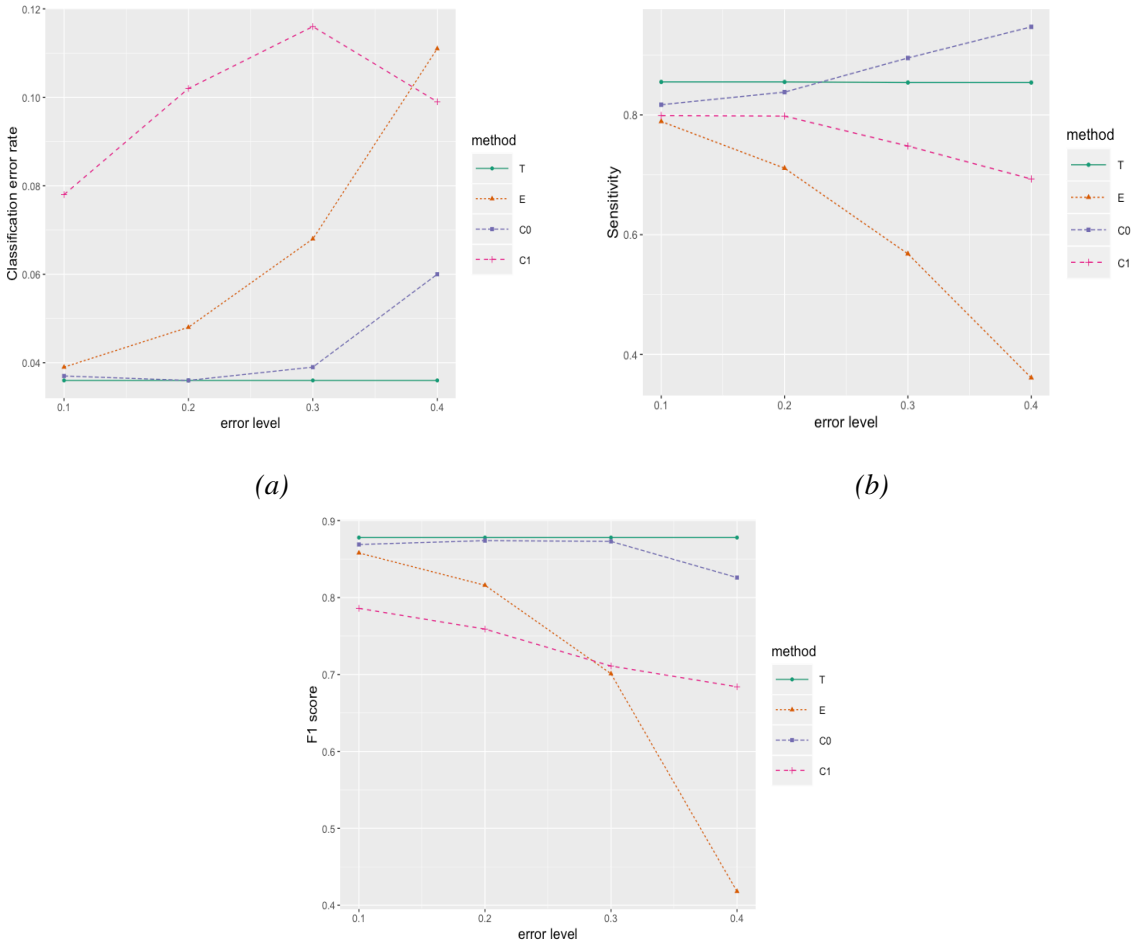


Figure 2.2: Simulation results for logistic regression with class 1 proportion ϕ=15%, sample size being 5000 and validation size being 200. The misclassification error depends linearly on the covariates.

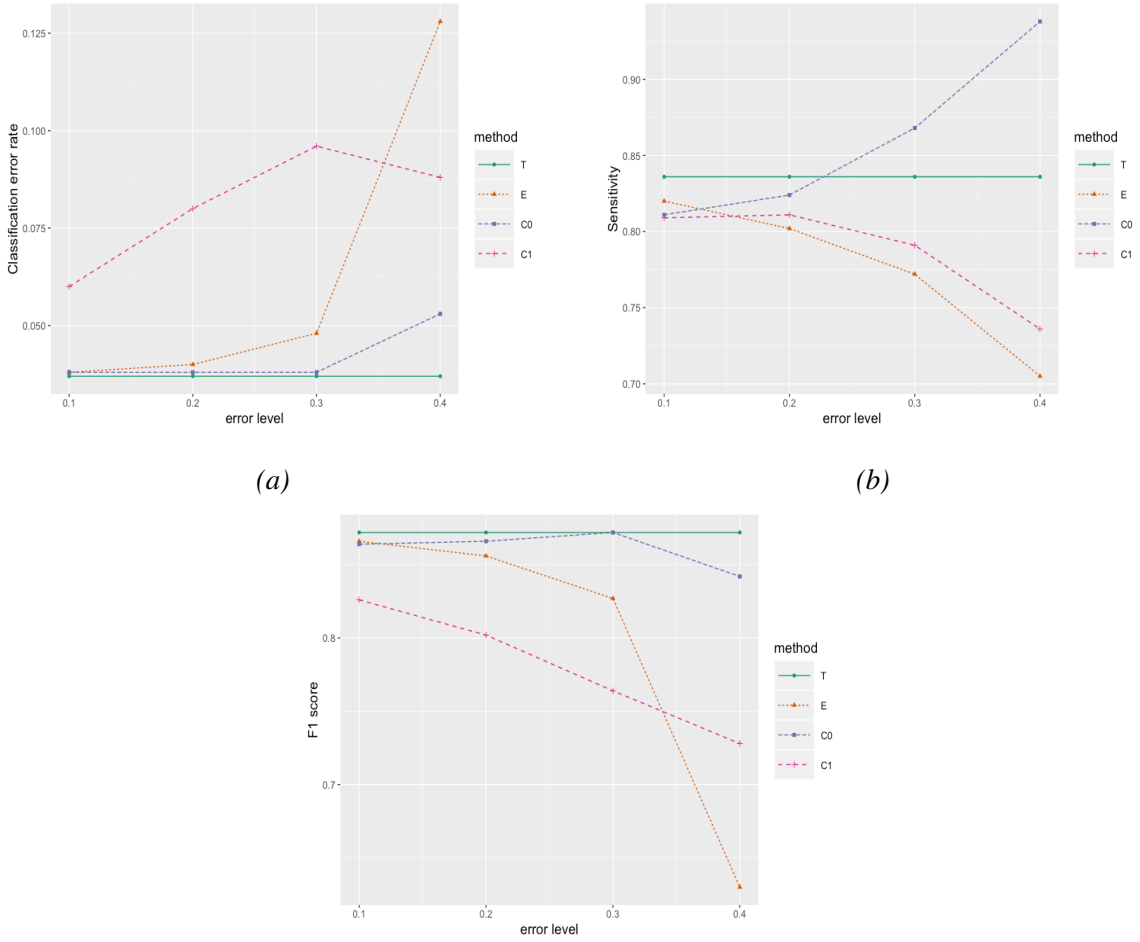


Figure 2.3: Simulation results for SVM with class 1 proportion ϕ=15%, sample size being 5000 and validation size being 200. The misclassification error depends linearly on the covariates.

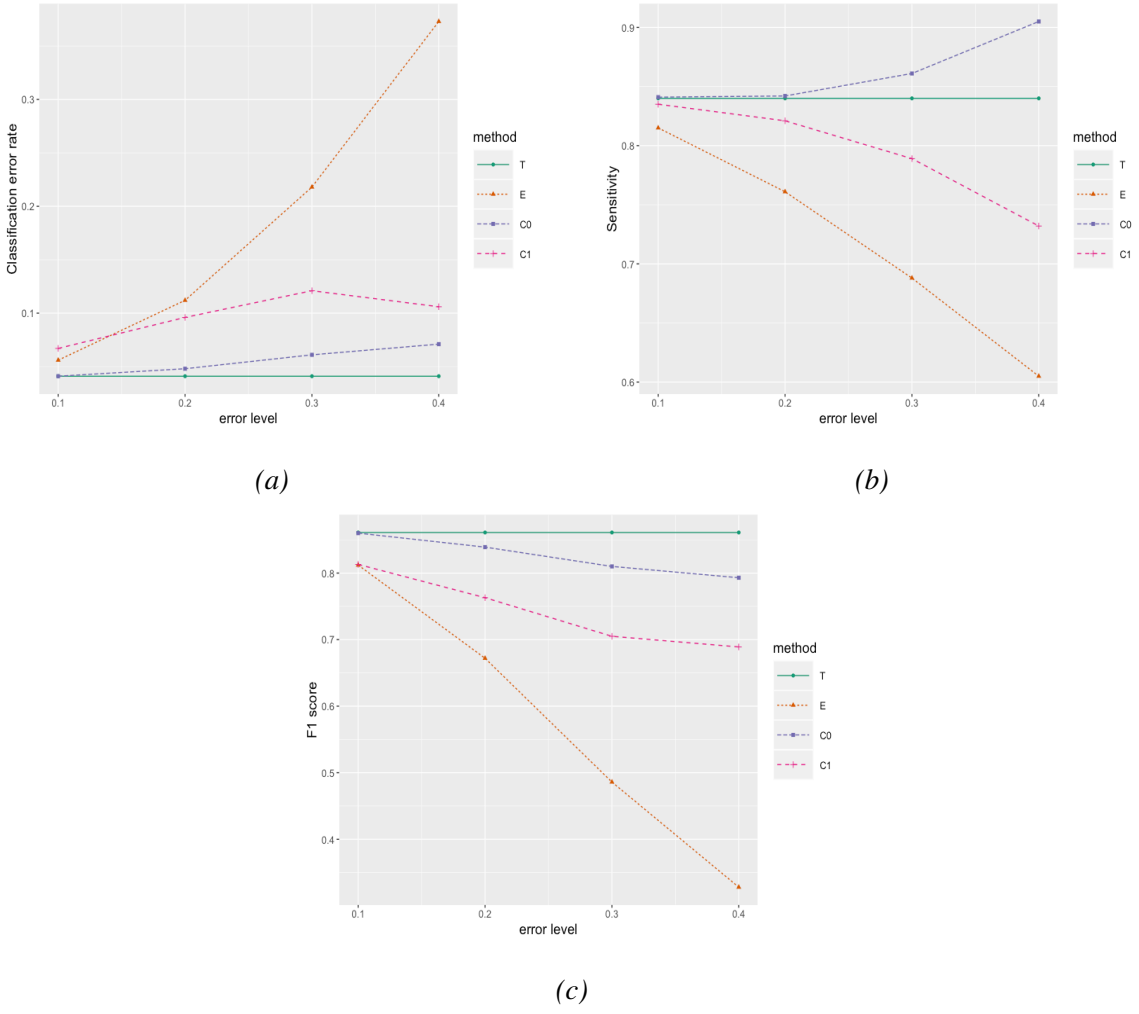


Figure 2.4: Simulation results for random forest with class 1 proportion ϕ=15%, sample size being 5000 and validation size being 200. The misclassification error depends linearly on the covariates.

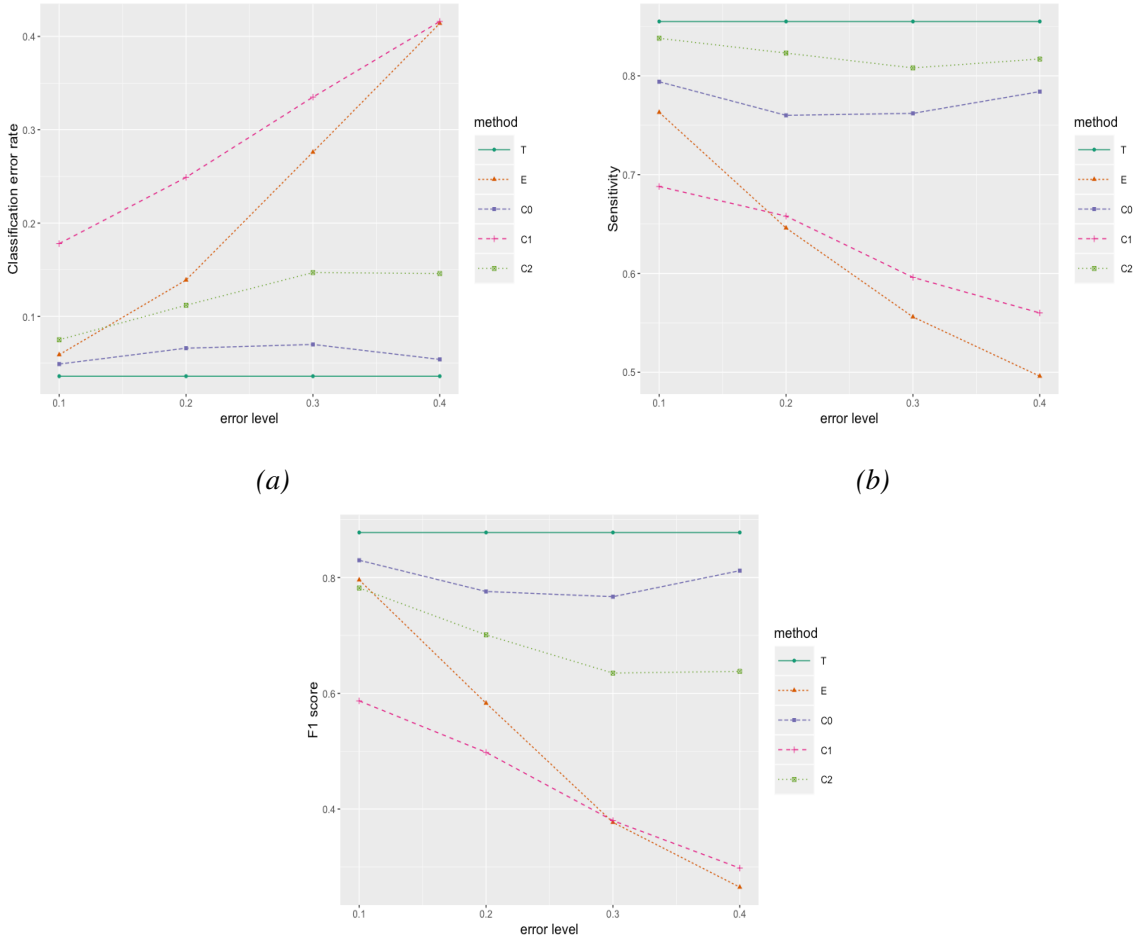
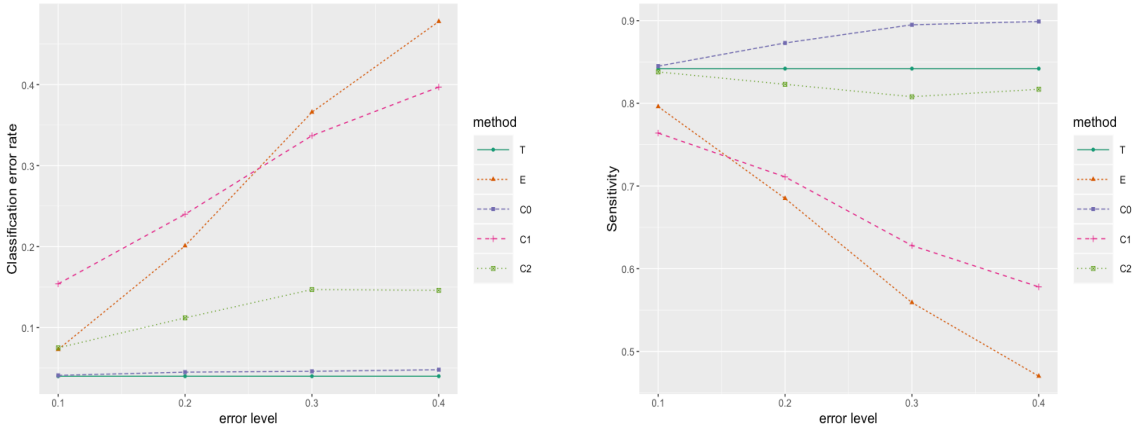
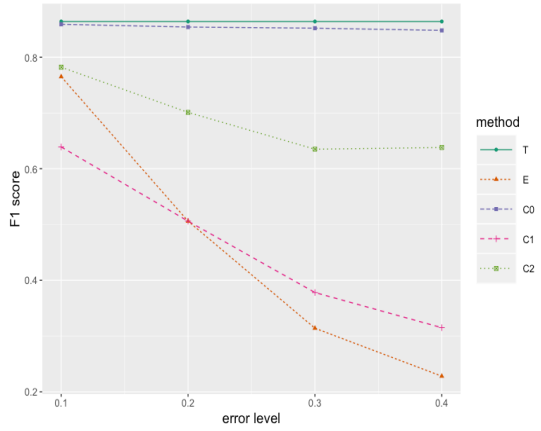


Figure 2.5: Simulation results for logistic regression with class 1 proportion ϕ=15%, sample size being 5000 and validation size being 200. The misclassification error depends nonlinearly on the covariates.



(a)

(b)



(c)

Figure 2.6: Simulation results for KNN with class 1 proportion ϕ=15%, sample size being 5000 and validation size being 200. The misclassification error depends nonlinearly on the covariates.

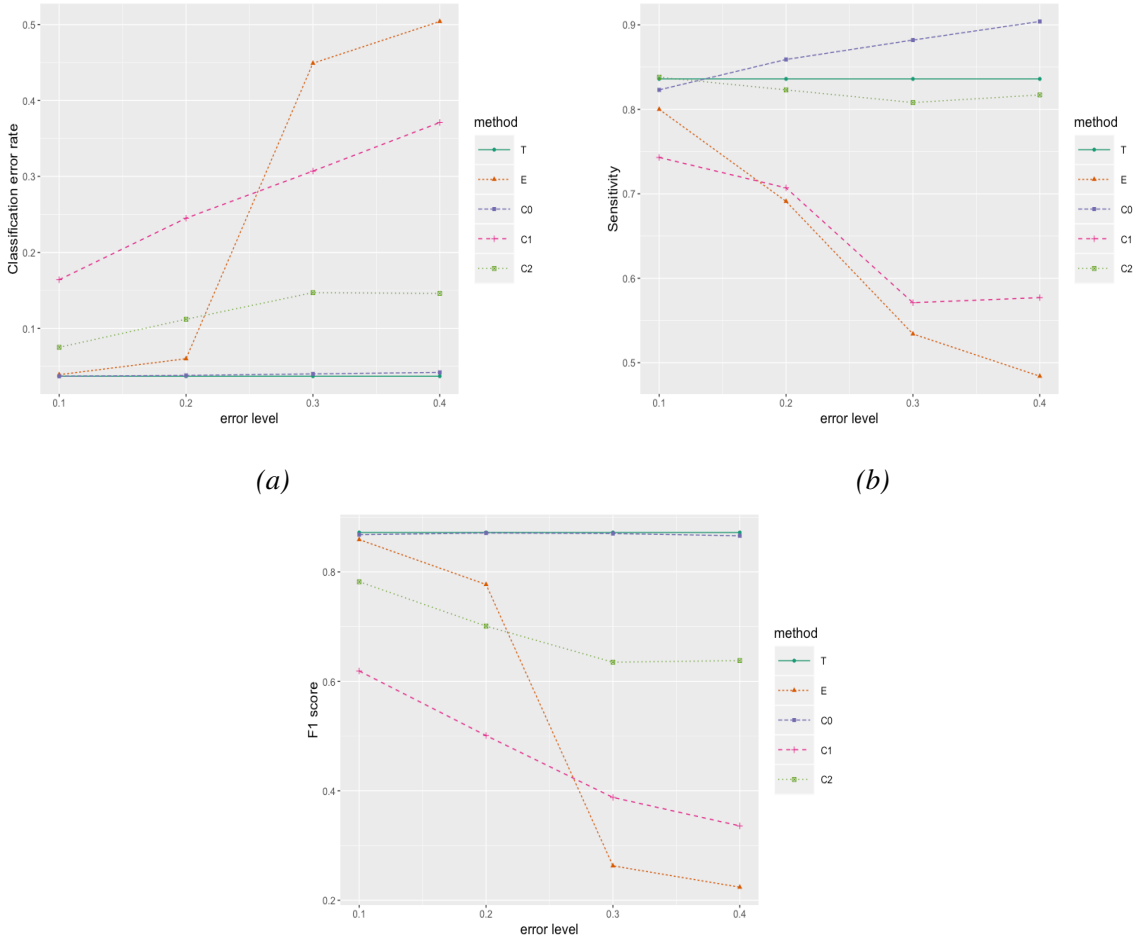
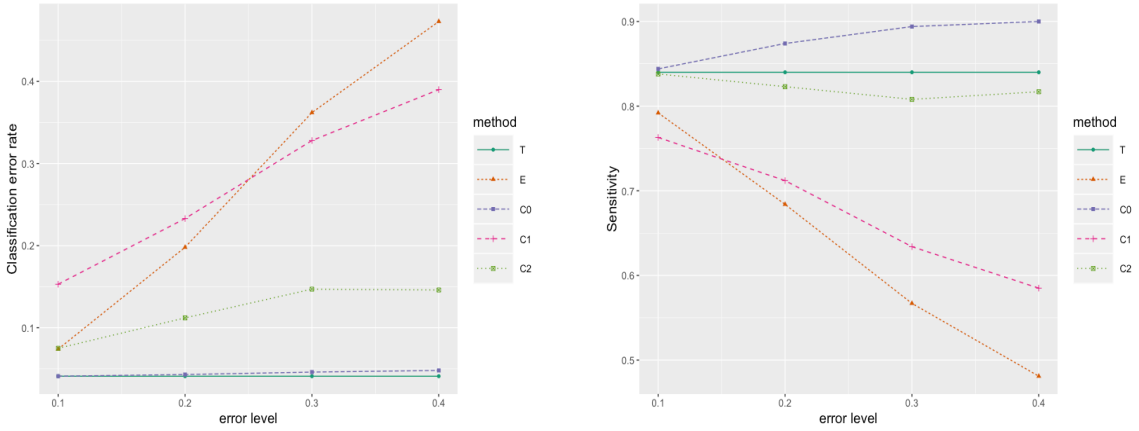
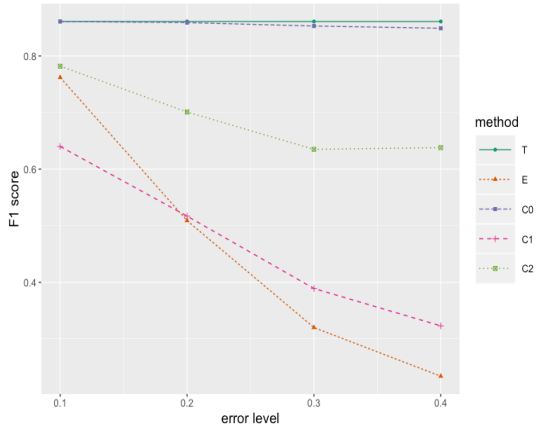


Figure 2.7: Simulation results for SVM with class 1 proportion ϕ=15%, sample size being 5000 and validation size being 200. The misclassification error depends nonlinearly on the covariates.



(a)

(b)



(c)

Figure 2.8: Simulation results for random forest with class 1 proportion ϕ=15%, sample size being 5000 and validation size being 200. The misclassification error depends nonlinearly on the covariates.

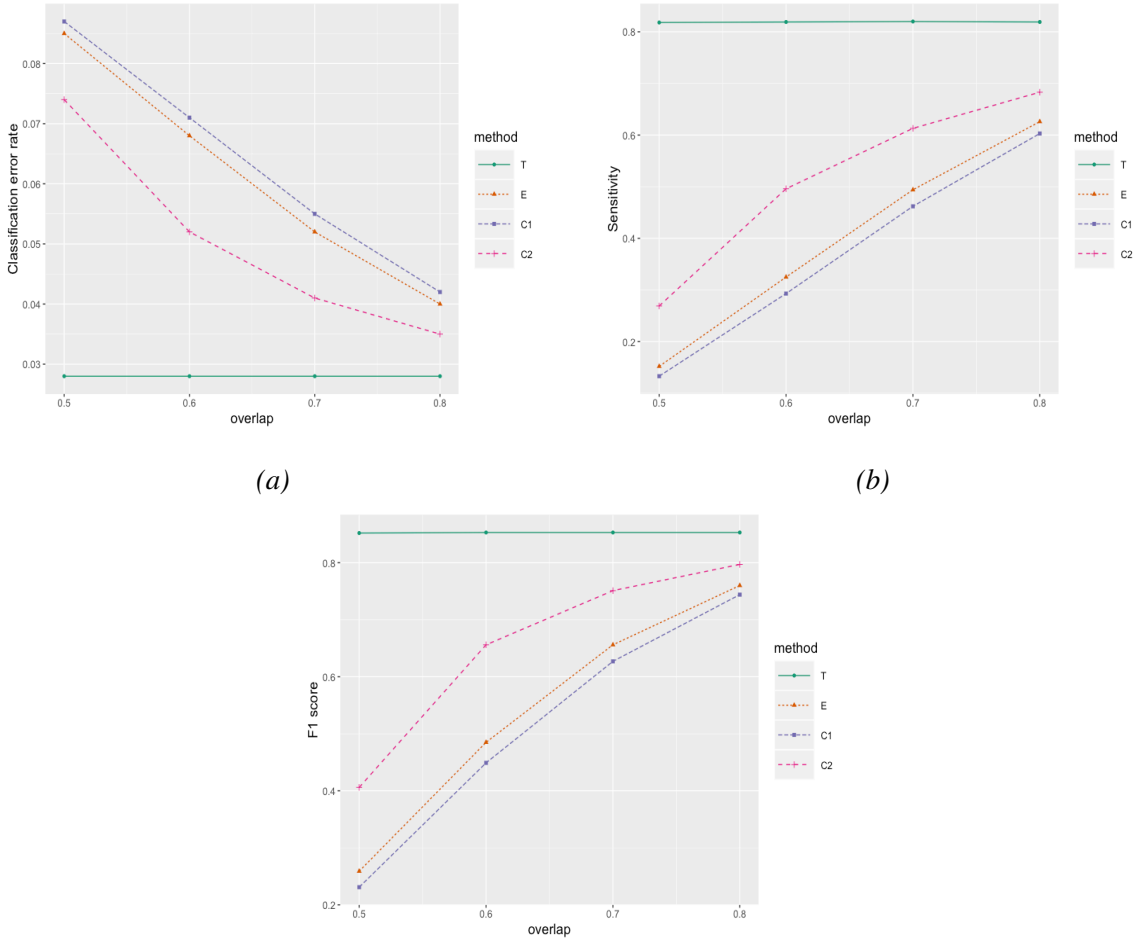
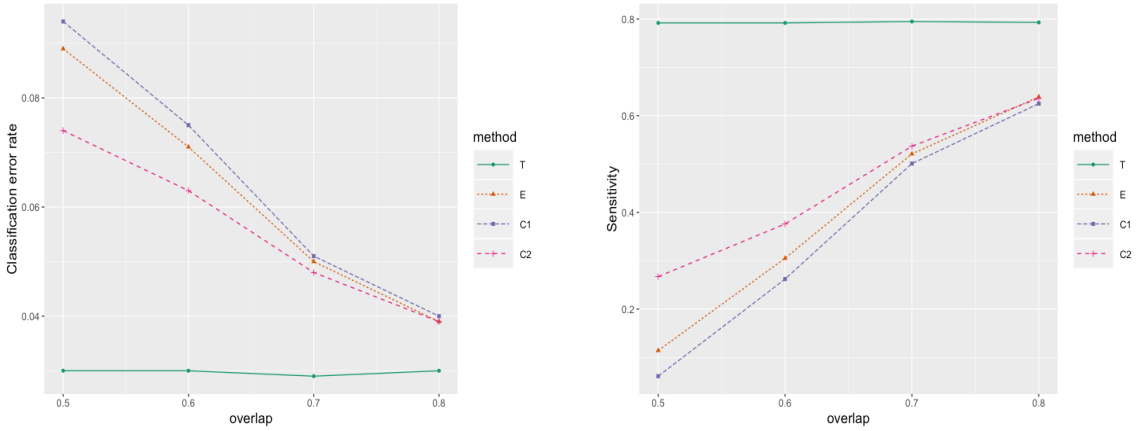
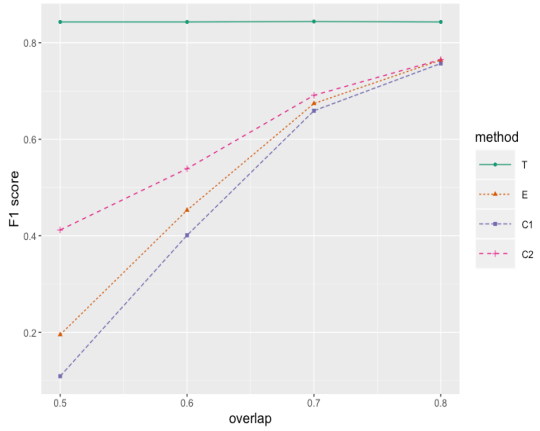


Figure 2.9: Simulation results for logistic regression with class 1 proportion ϕ=10%, and sample size being 5000. The misclassification error is caused by shifting the class 1 data by a distance.



(a)

(b)



(c)

Figure 2.10: Simulation results for SVM with class 1 proportion ϕ=10%, sample size being 5000. The misclassification error is caused by shifting the class 1 data by a distance.

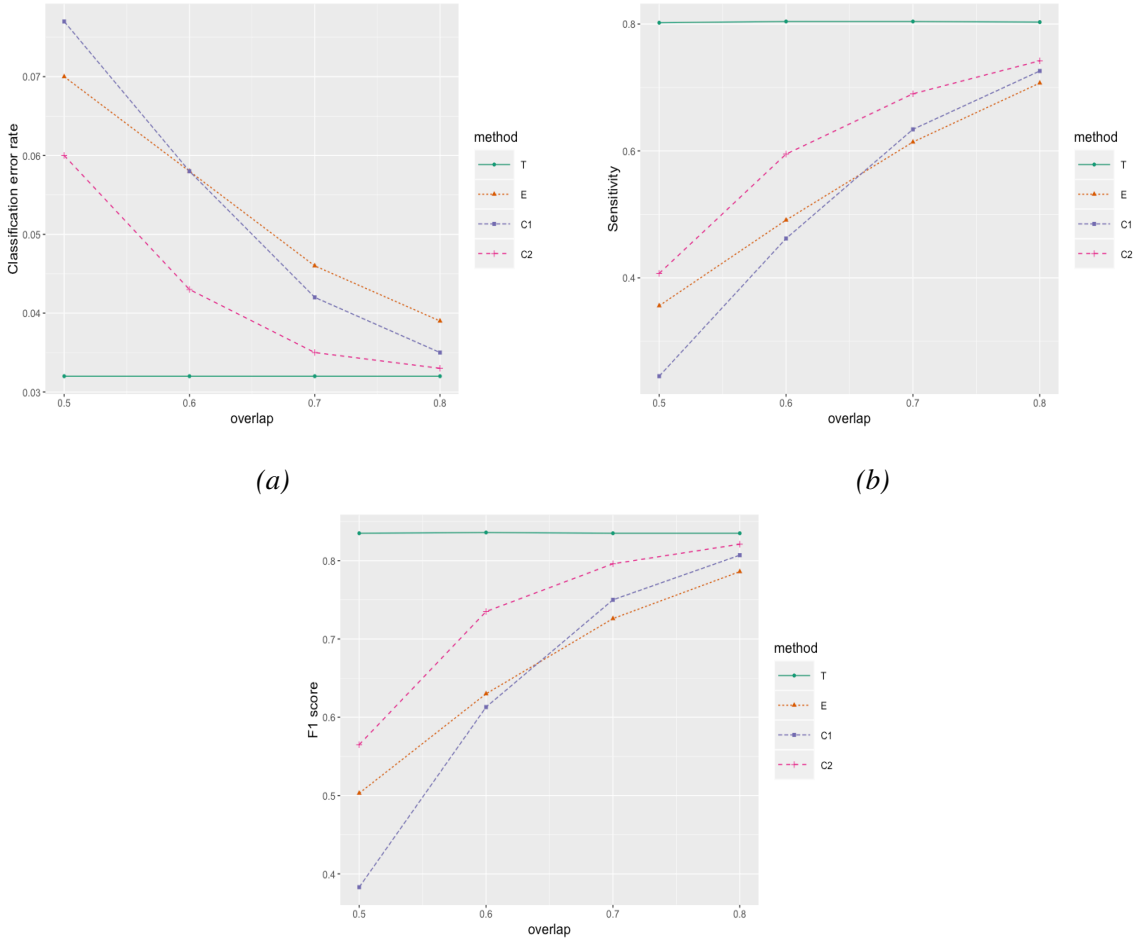


Figure 2.11: Simulation results for KNN with class 1 proportion ϕ=10%, sample size being 5000. The misclassification error is caused by shifting the class 1 data by a distance.

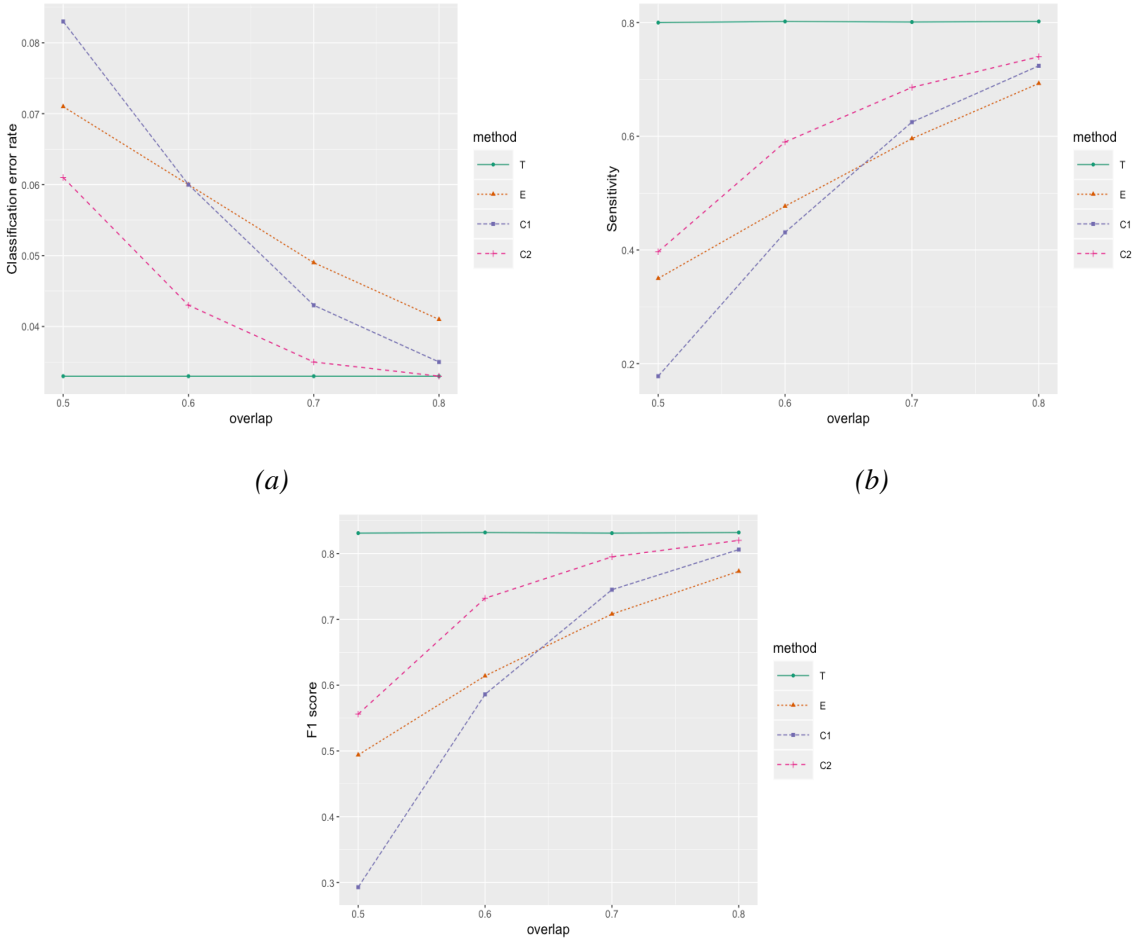


Figure 2.12: Simulation results for random forest classifier with class 1 proportion ϕ=10%, sample size being 5000. The misclassification error is caused by shifting the class 1 data by a distance.

Table 2.1: Simulation results for KNN classifier with linear $\gamma_{01}(Z)$ and $\gamma_{10}(Z)$. The data size is 5000, class 1 proportion is 15%, and the validation size for C1 is 200.

			T		
error level	classification error rate	sensitivity	specificity	F1	G score
0.1	0.040(0.004)	0.842(0.022)	0.981(0.004)	0.864(0.014)	0.864(0.014)
0.2	0.040(0.004)	0.842(0.022)	0.981(0.004)	0.864(0.014)	0.864(0.014)
0.3	0.040(0.004)	0.842(0.022)	0.981(0.004)	0.864(0.014)	0.864(0.014)
0.4	0.040(0.004)	0.842(0.022)	0.981(0.004)	0.864(0.014)	0.864(0.014)
			E		
error level	classification error rate	sensitivity	specificity	F1	G score
0.1	0.053(0.006)	0.822(0.026)	0.970(0.006)	0.824(0.018)	0.825(0.018)
0.2	0.109(0.011)	0.773(0.035)	0.912(0.013)	0.681(0.028)	0.687(0.028)
0.3	0.218(0.019)	0.701(0.047)	0.796(0.024)	0.491(0.031)	0.515(0.030)
0.4	0.374(0.027)	0.615(0.061)	0.628(0.034)	0.331(0.028)	0.373(0.031)
			C0		
error level	classification error rate	sensitivity	specificity	F1	G score
0.1	0.041(0.004)	0.838(0.024)	0.980(0.004)	0.859(0.015)	0.859(0.015)
0.2	0.052(0.006)	0.841(0.026)	0.967(0.007)	0.829(0.019)	0.829(0.019)
0.3	0.069(0.009)	0.859(0.026)	0.944(0.011)	0.790(0.024)	0.793(0.023)
0.4	0.078(0.014)	0.902(0.023)	0.925(0.018)	0.777(0.030)	0.785(0.026)
			C1		
error level	classification error rate	sensitivity	specificity	F1	G score
0.1	0.067(0.086)	0.835(0.094)	0.951(0.104)	0.813(0.027)	0.819(0.010)
0.2	0.099(0.128)	0.821(0.124)	0.916(0.154)	0.756(0.158)	0.766(0.138)
0.3	0.127(0.141)	0.788(0.160)	0.888(0.165)	0.691(0.175)	0.705(0.158)
0.4	0.112(0.108)	0.734(0.206)	0.915(0.131)	0.679(0.181)	0.693(0.167)

Table 2.2: Simulation results for KNN classifier with linear $\gamma_{01}(Z)$ and $\gamma_{10}(Z)$. The data size is 1000, class 1 proportion is 15%, and the validation size for C1 is 200.

T					
error level	classification error rate	sensitivity	specificity	F1	G score
0.1	0.041(0.009)	0.827(0.052)	0.982(0.008)	0.857(0.033)	0.858(0.032)
0.2	0.041(0.009)	0.827(0.052)	0.982(0.008)	0.857(0.033)	0.858(0.032)
0.3	0.041(0.009)	0.827(0.052)	0.982(0.008)	0.857(0.033)	0.858(0.032)
0.4	0.041(0.009)	0.827(0.052)	0.982(0.008)	0.857(0.033)	0.858(0.032)
E					
error level	classification error rate	sensitivity	specificity	F1	G score
0.1	0.055(0.013)	0.804(0.059)	0.970(0.013)	0.814(0.042)	0.816(0.042)
0.2	0.111(0.023)	0.755(0.077)	0.913(0.026)	0.672(0.060)	0.678(0.059)
0.3	0.220(0.038)	0.682(0.103)	0.797(0.043)	0.483(0.069)	0.505(0.070)
0.4	0.379(0.048)	0.592(0.121)	0.626(0.055)	0.319(0.060)	0.359(0.067)
C0					
error level	classification error rate	sensitivity	specificity	F1	G score
0.1	0.044(0.010)	0.814(0.057)	0.982(0.009)	0.848(0.036)	0.849(0.035)
0.2	0.055(0.014)	0.817(0.059)	0.968(0.015)	0.818(0.044)	0.819(0.043)
0.3	0.071(0.020)	0.839(0.059)	0.945(0.024)	0.782(0.053)	0.785(0.050)
0.4	0.080(0.024)	0.891(0.049)	0.926(0.030)	0.773(0.055)	0.782(0.049)
C1					
error level	classification error rate	sensitivity	specificity	F1	G score
0.1	0.087(0.132)	0.815(0.102)	0.931(0.158)	0.788(0.149)	0.798(0.126)
0.2	0.098(0.124)	0.807(0.119)	0.919(0.149)	0.752(0.149)	0.763(0.130)
0.3	0.114(0.107)	0.770(0.171)	0.907(0.128)	0.694(0.166)	0.706(0.152)
0.4	0.114(0.108)	0.721(0.209)	0.916(0.132)	0.671(0.183)	0.687(0.169)

Table 2.3: Simulation results for KNN classifier with nonlinear $\gamma_{01}(Z)$ and $\gamma_{10}(Z)$ and $\epsilon = 0.3$. The data size is 5000, and the validation size for C1 and C2 is 200.

T					
proportion	classification error rate	sensitivity	specificity	F1	G score
0.20	0.047(0.004)	0.866(0.018)	0.975(0.004)	0.881(0.011)	0.882(0.011)
0.15	0.040(0.004)	0.842(0.022)	0.981(0.004)	0.864(0.014)	0.864(0.014)
0.10	0.032(0.004)	0.803(0.030)	0.987(0.003)	0.835(0.019)	0.836(0.019)
E					
proportion	classification error rate	sensitivity	specificity	F1	G score
0.20	0.372(0.016)	0.574(0.034)	0.642(0.018)	0.382(0.021)	0.405(0.022)
0.15	0.366(0.016)	0.559(0.038)	0.647(0.017)	0.314(0.022)	0.350(0.023)
0.10	0.361(0.016)	0.533(0.048)	0.651(0.017)	0.228(0.021)	0.278(0.025)
C0					
proportion	classification error rate	sensitivity	specificity	F1	G score
0.20	0.050(0.004)	0.908(0.016)	0.960(0.005)	0.879(0.010)	0.879(0.010)
0.15	0.046(0.004)	0.895(0.019)	0.964(0.005)	0.852(0.013)	0.854(0.013)
0.10	0.042(0.004)	0.873(0.025)	0.967(0.004)	0.806(0.018)	0.808(0.018)
C1					
proportion	classification error rate	sensitivity	specificity	F1	G score
0.20	0.334(0.139)	0.633(0.165)	0.674(0.178)	0.445(0.122)	0.470(0.117)
0.15	0.337(0.144)	0.628(0.160)	0.669(0.175)	0.378(0.113)	0.415(0.106)
0.10	0.340(0.163)	0.620(0.153)	0.664(0.183)	0.295(0.109)	0.347(0.100)
C2					
proportion	classification error rate	sensitivity	specificity	F1	G score
0.20	0.147(0.057)	0.819(0.065)	0.861(0.073)	0.697(0.072)	0.707(0.064)
0.15	0.147(0.064)	0.808(0.060)	0.861(0.078)	0.635(0.083)	0.653(0.071)
0.10	0.142(0.068)	0.790(0.070)	0.865(0.077)	0.545(0.096)	0.575(0.080)

Table 2.4: Simulation results for different scenarios with random forest classifier and sample size being 5000. The misclassification error is caused by shifting the class 1 data by a distance.

		T			
	class 1 proportion	classification error rate	sensitivity	F1 score	G score
	0.10	0.033(0.004)	0.800(0.030)	0.831(0.020)	0.832(0.019)
	0.15	0.041(0.004)	0.839(0.024)	0.860(0.014)	0.860(0.014)
	0.20	0.047(0.004)	0.866(0.018)	0.880(0.011)	0.880(0.011)
		E			
overlap	class 1 proportion	classification error rate	sensitivity	F1 score	G score
50%	0.10	0.071(0.007)	0.350(0.058)	0.494(0.061)	0.545(0.052)
	0.15	0.103(0.009)	0.380(0.050)	0.523(0.051)	0.566(0.043)
	0.20	0.136(0.010)	0.397(0.044)	0.538(0.043)	0.576(0.037)
60%	0.10	0.060(0.007)	0.477(0.059)	0.614(0.052)	0.644(0.044)
	0.15	0.084(0.008)	0.515(0.051)	0.646(0.042)	0.669(0.036)
	0.20	0.108(0.010)	0.540(0.045)	0.665(0.036)	0.685(0.032)
70%	0.10	0.049(0.006)	0.596(0.052)	0.708(0.039)	0.722(0.035)
	0.15	0.066(0.007)	0.640(0.043)	0.742(0.031)	0.752(0.028)
	0.20	0.082(0.008)	0.672(0.038)	0.765(0.026)	0.773(0.024)
80%	0.10	0.041(0.005)	0.693(0.043)	0.773(0.029)	0.779(0.027)
	0.15	0.053(0.006)	0.739(0.035)	0.806(0.022)	0.810(0.021)
	0.20	0.064(0.006)	0.770(0.030)	0.828(0.019)	0.831(0.018)
		C1			
overlap	class 1 proportion	classification error rate	sensitivity	F1 score	G score
50%	0.10	0.083(0.009)	0.178(0.081)	0.293(0.114)	0.402(0.098)
	0.15	0.120(0.013)	0.211(0.077)	0.339(0.102)	0.442(0.084)
	0.20	0.155(0.016)	0.236(0.074)	0.373(0.095)	0.460(0.070)
60%	0.10	0.060(0.010)	0.431(0.097)	0.586(0.095)	0.634(0.074)
	0.15	0.080(0.012)	0.496(0.084)	0.647(0.074)	0.683(0.059)
	0.20	0.098(0.015)	0.542(0.076)	0.687(0.063)	0.659(0.057)
70%	0.10	0.043(0.006)	0.625(0.065)	0.745(0.047)	0.761(0.040)
	0.15	0.054(0.008)	0.691(0.053)	0.793(0.035)	0.803(0.031)
	0.20	0.062(0.008)	0.740(0.043)	0.826(0.026)	0.785(0.035)
80%	0.10	0.035(0.005)	0.724(0.045)	0.806(0.028)	0.812(0.025)
	0.15	0.043(0.005)	0.780(0.033)	0.843(0.019)	0.846(0.017)
	0.20	0.050(0.005)	0.817(0.027)	0.867(0.015)	0.844(0.020)
		C2			
overlap	class 1 proportion	classification error rate	sensitivity	F1 score	G score
50%	0.10	0.061(0.012)	0.397(0.118)	0.556(0.130)	0.618(0.104)
	0.15	0.080(0.016)	0.473(0.103)	0.633(0.100)	0.680(0.079)
	0.20	0.094(0.018)	0.539(0.087)	0.693(0.076)	0.728(0.061)
60%	0.10	0.043(0.007)	0.590(0.065)	0.732(0.051)	0.755(0.041)
	0.15	0.054(0.007)	0.662(0.047)	0.786(0.032)	0.801(0.027)
	0.20	0.062(0.008)	0.710(0.041)	0.820(0.026)	0.831(0.022)
70%	0.10	0.035(0.005)	0.686(0.045)	0.795(0.029)	0.805(0.025)
	0.15	0.044(0.005)	0.746(0.033)	0.835(0.020)	0.842(0.018)
	0.20	0.050(0.005)	0.788(0.027)	0.862(0.015)	0.866(0.014)
80%	0.10	0.033(0.004)	0.740(0.037)	0.820(0.023)	0.825(0.021)
	0.15	0.040(0.004)	0.793(0.028)	0.855(0.016)	0.857(0.015)
	0.20	0.046(0.005)	0.827(0.023)	0.877(0.013)	0.879(0.012)

Table 2.5: The effect of sample size on the proposed method for KNN classifier with class 1 proportion being 15% and overlap proportion being 70%.

sample size	methods	classification error rate	sensitivity	specificity	F1 score	G score
1000	T	0.040(0.009)	0.828(0.052)	0.983(0.007)	0.859(0.032)	0.860(0.032)
	E	0.065(0.016)	0.636(0.102)	0.988(0.007)	0.740(0.076)	0.754(0.067)
	C1	0.060(0.017)	0.642(0.118)	0.992(0.006)	0.755(0.089)	0.772(0.075)
	C2	0.057(0.015)	0.656(0.102)	0.994(0.005)	0.771(0.075)	0.787(0.063)
5000	T	0.040(0.004)	0.843(0.023)	0.981(0.004)	0.864(0.015)	0.865(0.014)
	E	0.063(0.007)	0.654(0.044)	0.987(0.003)	0.755(0.031)	0.765(0.028)
	C1	0.054(0.008)	0.688(0.052)	0.991(0.003)	0.790(0.035)	0.800(0.031)
	C2	0.047(0.005)	0.732(0.036)	0.992(0.003)	0.824(0.023)	0.831(0.020)

Table 2.6: Simulation results for the robustness of the proposed method on random forest classifier with class 1 proportion being 15% and sample size being 5000. The data is error-free and is corrected with the proposed method for different overlap proportions.

			T		
	classification error rate	sensitivity	specificity	F1	G score
	0.041(0.004)	0.840(0.023)	0.980(0.004)	0.860(0.014)	0.860(0.014)
			C1		
overlap	classification error rate	sensitivity	specificity	F1	G score
50%	0.046(0.006)	0.758(0.042)	0.988(0.003)	0.830(0.024)	0.834(0.022)
60%	0.044(0.005)	0.786(0.033)	0.986(0.004)	0.843(0.018)	0.846(0.017)
70%	0.042(0.004)	0.806(0.030)	0.985(0.004)	0.851(0.017)	0.852(0.016)
80%	0.042(0.004)	0.819(0.027)	0.983(0.004)	0.854(0.016)	0.855(0.015)
			C2		
overlap	classification error rate	sensitivity	specificity	F1	G score
50%	0.067(0.011)	0.561(0.071)	0.998(0.001)	0.712(0.058)	0.741(0.047)
60%	0.050(0.006)	0.695(0.040)	0.995(0.002)	0.807(0.026)	0.818(0.022)
70%	0.043(0.005)	0.759(0.031)	0.992(0.002)	0.841(0.018)	0.846(0.017)
80%	0.041(0.004)	0.798(0.028)	0.988(0.003)	0.855(0.016)	0.857(0.015)

Table 2.7: Classification results for patient 1015 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.076	0.082	0.100	0.086	0.137	0.084	0.083	0.077
	sensitivity	0.745	0.782	0.615	0.671	0.595	0.635	0.565	0.610
	specificity	0.946	0.935	0.935	0.944	0.895	0.949	0.961	0.962
	F1 score	0.682	0.677	0.573	0.630	0.486	0.621	0.599	0.635
1.39	classification error	0.089	0.092	0.114	0.102	0.151	0.104	0.102	0.101
	sensitivity	0.669	0.694	0.550	0.549	0.542	0.588	0.480	0.515
	specificity	0.947	0.940	0.936	0.944	0.896	0.948	0.961	0.957
	F1 score	0.663	0.664	0.558	0.601	0.486	0.580	0.552	0.573
2.33	classification error	0.065	0.067	0.089	0.071	0.129	0.072	0.071	0.060
	sensitivity	0.855	0.890	0.691	0.739	0.661	0.700	0.638	0.704
	specificity	0.944	0.937	0.934	0.949	0.893	0.951	0.959	0.964
	F1 score	0.712	0.713	0.593	0.661	0.490	0.645	0.627	0.687

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 2.8: Classification results for patient 2008 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.089	0.087	0.141	0.063	0.069	0.049	0.116	0.069
	sensitivity	0.000	0.000	0.400	0.432	0.621	0.575	0.304	0.266
	specificity	0.998	1.000	0.903	0.985	0.961	0.986	0.940	0.994
	F1 score	0.000	0.000	0.330	0.545	0.612	0.670	0.314	0.401
1.39	classification error	0.104	0.242	0.144	0.087	0.077	0.072	0.123	0.089
	sensitivity	0.000	0.000	0.398	0.377	0.550	0.530	0.301	0.281
	specificity	0.999	0.845	0.908	0.974	0.965	0.973	0.942	0.983
	F1 score	0.000	0.001	0.362	0.469	0.594	0.600	0.334	0.392
2.33	classification error	0.083	0.082	0.141	0.055	0.065	0.037	0.116	0.053
	sensitivity	0.000	0.000	0.375	0.473	0.638	0.616	0.278	0.386
	specificity	0.998	1.000	0.902	0.986	0.961	0.994	0.938	0.997
	F1 score	0.000	0.000	0.303	0.582	0.615	0.730	0.281	0.544

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 2.9: Classification results for patient 1012 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.091	0.026	0.096	0.024	0.144	0.017	0.054	0.015
	sensitivity	0.754	0.307	0.834	0.688	0.794	0.688	0.794	0.769
	specificity	0.911	0.985	0.905	0.981	0.858	0.987	0.948	0.988
	F1 score	0.210	0.275	0.219	0.486	0.152	0.560	0.321	0.400
1.39	classification error	0.095	0.035	0.099	0.027	0.145	0.018	0.057	0.021
	sensitivity	0.691	0.303	0.763	0.723	0.763	0.773	0.766	0.727
	specificity	0.911	0.982	0.905	0.979	0.857	0.987	0.947	0.985
	F1 score	0.264	0.301	0.275	0.566	0.205	0.679	0.399	0.630
2.33	classification error	0.089	0.018	0.096	0.019	0.144	0.014	0.054	0.009
	sensitivity	0.861	0.267	0.960	0.881	0.881	0.911	0.960	0.941
	specificity	0.911	0.987	0.904	0.982	0.856	0.987	0.946	0.991
	F1 score	0.137	0.194	0.142	0.440	0.092	0.517	0.228	0.617

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 2.10: Classification results for patient 1035 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.009	0.011	0.108	0.029	0.107	0.014	0.060	0.021
	sensitivity	0.653	0.173	0.918	0.959	0.643	0.765	0.643	0.990
	specificity	0.995	0.998	0.892	0.972	0.895	0.989	0.943	0.979
	F1 score	0.637	0.260	0.163	0.434	0.120	0.562	0.196	0.519
1.39	classification error	0.009	0.011	0.098	0.050	0.100	0.027	0.057	0.037
	sensitivity	0.663	0.529	0.817	0.885	0.538	0.750	0.519	0.885
	specificity	0.995	0.994	0.903	0.951	0.905	0.976	0.948	0.964
	F1 score	0.654	0.539	0.173	0.309	0.119	0.411	0.184	0.372
2.33	classification error	0.021	0.023	0.085	0.045	0.134	0.028	0.052	0.043
	sensitivity	0.367	0.480	0.803	0.834	0.755	0.742	0.537	0.699
	specificity	0.997	0.992	0.919	0.958	0.870	0.979	0.959	0.964
	F1 score	0.499	0.543	0.347	0.506	0.240	0.596	0.364	0.473

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 2.11: Classification results for patient 2009 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.055	0.053	0.057	0.044	0.037	0.034	0.045	0.036
	sensitivity	0.000	0.000	0.747	0.783	0.769	0.726	0.675	0.715
	specificity	0.984	0.984	0.951	0.963	0.971	0.976	0.966	0.974
	F1 score	0.000	0.000	0.503	0.581	0.621	0.626	0.537	0.609
1.39	classification error	0.063	0.062	0.061	0.050	0.039	0.036	0.047	0.041
	sensitivity	0.000	0.000	0.735	0.743	0.749	0.705	0.678	0.673
	specificity	0.982	0.984	0.949	0.960	0.971	0.976	0.966	0.973
	F1 score	0.000	0.000	0.529	0.579	0.639	0.642	0.571	0.603
2.33	classification error	0.048	0.049	0.053	0.042	0.033	0.030	0.041	0.031
	sensitivity	0.000	0.000	0.770	0.805	0.787	0.758	0.696	0.760
	specificity	0.984	0.983	0.953	0.963	0.973	0.978	0.968	0.976
	F1 score	0.000	0.000	0.485	0.553	0.610	0.625	0.526	0.611

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Chapter 3

Weighted correction model

3.1 Introduction

In the previous chapter, we proposed a predict probability correction method and investigated its performance in different scenarios. We estimated the misclassification probabilities by the position of each data point. In this chapter, we further extend this idea and propose a weighted model method for different classifiers. This weighted model method incorporates the weight for each data point in the model building process. A point with a larger weight has more influence in the model parameter estimation. The numerical study shows the proposed method has great performance when misclassification appears in response.

The rest of the section organizes as follows. The proposed weighted models for different classifiers are introduced in section 3.2. In section 3.3 simulation studies and the application to the prostate cancer imaging data are carried out to test the performance of the proposed method. Conclusions are made in 3.4.

3.2 Framework and Method description

Let $Y = \{0, 1\}$ denote the true binary response that may not be directly observed, and $\boldsymbol{Z}$ the covariate vector that is error-free with dimension p. The observed version of Y is Y^*.

In Chapter 2.3.2, the weight calculation was proposed for each data point. The weight ω is defined as the proportion of points in the circle that have the same class labels as the centre point. The circle represents the possible area of the true response for the centre point, so the weight can be viewed as an estimate of the probability that measures how likely the true response Y is equal to the observed Y^*: $\omega_i \approx \Pr(Y_i^* = Y_i)$, $i = 1, \ldots, n$. The weight reflects the importance of each data point, which inspires us to fit weighted models to the data.

3.2.1 Weighted logistic regression

The weighted likelihood for logistic regression can be written as

$$L^*(\boldsymbol{\theta}) = \sum_{i=1}^{n} g(\omega_i) l_i(\boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ is the regression coefficient to be estimated, $l_i(\boldsymbol{\theta})$ is the original likelihood for each data point without weight, and $g(\omega_i)$ is a data-adaptive weight function. The detailed calculation of $l_i(\boldsymbol{\theta})$ can be found in 1.2.1.

The function g can be the identity function, in this case the weighted likelihood becomes

$$L^*(\boldsymbol{\theta}) = \sum_{i=1}^{n} \omega_i l_i(\boldsymbol{\theta}),$$

which is the traditional weighted logistic regression. In the simulation study, we have found that using a different weight function rather than the identity function could lead to better fitting results. For example, let

$$g(\omega_i) = \begin{cases} \omega_i & \text{if } \omega_i \geq w, \\ 0 & \text{if } \omega_i < w, \end{cases}$$

where w is chosen so that only points with weights greater than or equal to w will be used in the model fitting. If $w = 1$, then only the points that have probability 1 that the true response equals to the observed response are used for fitting, which is the same as the preliminary model m_1 discussed in the new weight calculation in 2.3.2. On the other hand, if $w = 0$, then all points are used, which corresponds to the traditional weighted logistic regression. The value of w can be decided using cross-validation.

3.2.2 Weighted SVM

Yang et al. (2007) proposed a weighted support vector machine method.

The original SVM can be written as a quadratic programming problem

$$\begin{aligned} L_D &= \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \langle h(\boldsymbol{z}_i), h(\boldsymbol{z}_j) \rangle \\ &= \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\boldsymbol{z}_i, \boldsymbol{z}_j) \end{aligned}$$

subject to

$$\sum_{i=1}^{l} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l$$

where α is the Lagrangian parameter, C is the penalty parameter, and l is the training data size.

The weighted model proposed by Yang et al. (2007) is

$$L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\boldsymbol{z_i}, \boldsymbol{z_j})$$

subject to

$$\sum_{i=1}^{l} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq \omega_i C, \quad i = 1, \ldots, l.$$

The basic idea of the weighted SVM is to put a weighted penalty for each data point in the training set. If a data point has a small weight, it means the observed response of this point is unlikely to be the same as the true response, so the penalty for misclassifying this point is small. Consequently, this data point will have a small influence in the estimation of the parameters. In this way, the influence of misclassification in response is reduced.

We can replace the weight ω by a function $g(\omega)$, then the restriction becomes:

$$\sum_{i=1}^{l} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq g(\omega_i) C, \quad i = 1, \ldots, l.$$

3.2.3 Weighted KNN

In the original KNN, the class of the new observation with K nearest points is decided by

$$\hat{y} = \text{argmax}_r \left\{ \sum_{j=1}^{K} I(y_j = r) \right\},$$

where $r = \{0, 1\}$ for binary class case. The weighted KNN model proposed by Hechenbichler and Schliep (2004) changes the decision rule to

$$\hat{y} = \text{argmax}_r \left\{ \sum_{j=1}^{K} \omega_i I(y_j = r) \right\}.$$

Similarly, we will modify it to

$$\hat{y} = \text{argmax}_r \left\{ \sum_{j=1}^{K} g(\omega_i) I(y_j = r) \right\}$$

to make it a data-adaptive weighted KNN model.

3.2.4 Weighted classification tree

In classification tree the node impurity measure serves as the criteria for splitting nodes and pruning the tree. In a node m of a classification tree, let R_m denote a region with N_m observations, then

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{z_i \in R_m} I(y_i = k)$$

is the observed proportion of class k in node m. The observations in node m are classified to class $k(m) = \arg\max_k \hat{p}_{mk}$, which is the majority class in node m. For an original classification tree, the commonly used node impurity measures are misclassification error, gini index, and cross-entropy or deviance (Friedman et al., 2001):

Misclassification error: $\frac{1}{N_m} \sum_{i \in R_m} I\{y_i \neq k(m)\}.$

Gini index: $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'}.$

Cross-entropy or deviance: $-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}.$

Usually gini index and cross-entropy are preferred.

In order to add weight for each observation in the classification tree, we propose the weighted impurity measures. Denote

$$\tilde{p}_{mk} = \frac{\sum_{z_i \in R_m} \omega_i I(y_i = k)}{\sum_{z_i \in R_m} \omega_i}$$

as the weighted observed proportion of class k in node m. Then the weighted impurity measures are:

Misclassification error:	$\frac{\sum_{z_i \in R_m} \omega_i I(y_i \neq k)}{\sum_{z_i \in R_m} \omega_i}$.
Gini index:	$\sum_{k \neq k'} \tilde{p}_{mk} \tilde{p}_{mk'}$.
Cross-entropy or deviance:	$-\sum_{k=1}^{K} \tilde{p}_{mk} \log \tilde{p}_{mk}$.

Substituting ω_i to $g(\omega_i)$ gives the data-adaptive weighted impurity measures.

3.3 Numerical investigation

Both simulation studies and real data application of the proposed method are presented in this section. The numerical studies were done using R 3.5.2 (R Core Team, 2018). The packages *e1071* (Meyer et al., 2019), *class* (Venables and Ripley, 2002) and *kernlab* (Karatzoglou et al., 2004) were used to perform the corresponding analysis using SVM, classification tree and KNN.

3.3.1 Simulation study

Simulation studies were carried out to test the performance of the weighted model method.

The data generation procedure was the same as described in Chapter 2.4.1. The cancer tissue, which was the set of data with class label 1, was approximated by a circle. The size and position of the circle were determined by the external source $\boldsymbol{W}$, in this case the Cartesian coordinates w_1 and w_2, i.e. $\boldsymbol{W} = (w_1, w_2)$. The class 1 proportion ϕ ranged among 0.1, 0.15, and

0.2 (for logistic regression ϕ being 0.05 was also tested), and the overlap proportion between the true cancer area and the observed cancer area was set to 0.5, 0.6, 0.7, and 0.8. The shift distance determined by the overlap proportion was assumed known.

The data of size 1000 or 5000 was randomly split into half training set and half testing set. In each fitting process the training set was used to train the model, and the testing set was used to test the performance by comparing the predicted class labels to the true class labels. The misclassification error rate, sensitivity, specificity, F1 score, and G score were recorded.

To evaluate the performance of the proposed weighted model for different classifiers, logistic regression, SVM, KNN and classification tree were considered in the simulation study. The number of nearest neighbors for KNN was set to 5. In splitting the classification tree, gini index was used. The maximum depth for splitting the tree was 10, and in the terminal node at least 5 data points were needed to make a decision. Radius kernel was considered for SVM, with gamma being 0.5, and the cost being 100.

The simulation procedure was as follows:

- step 1: generate the true data set (y_i, z_i), $i = 1, \ldots, n$;
- step 2: shift the circle of class 1 in the true data by a distance to create the error-prone data (y_i^*, z_i), $i = 1, \ldots, n$;
- step 3: calculate the raw weights with equation (2.4);
- step 4: update the raw weights with previously described method in Chapter 2.3.2. Logistic regression is served as the preliminary model to estimate the new weight (probability) in step 2.

The following scenarios were considered in the simulation:

- T: the classifier is trained and tested on the error-free data set (y_i, z_i), $i = 1, \ldots, n$.
- E: the classifier is trained and tested on the error-corrupted data set (y_i^*, z_i), $i = 1, \ldots, n$.

- W1: the proposed weighted classifier is trained and tested on the error-corrupted data set (y_i^*, z_i), $i = 1, \ldots, n$ with raw weights.

- W2: the proposed weighted classifier is trained and tested on the error-corrupted data set (y_i^*, z_i), $i = 1, \ldots, n$ with updated weights.

In W1 and W2 we considered to use the data-adaptive weight function $g(\omega)$ instead of directly using the weights ω. In this simulation study, we let

$$g(\omega_i) = \begin{cases} \omega_i & \text{if } \omega_i \geq w, \\ 0 & \text{if } \omega_i < w, \end{cases} \tag{3.1}$$

where w was a tuning parameter which was determined with 5-fold cross-validation.

As discussed in Chapter 2.4, the misclassification in response caused by the misalignment drastically decreased the classification performance. Small overlap proportion between y and y^* or highly imbalanced class proportions decreased the sensitivity, F1 score and G score significantly.

Figure 3.1 shows the F1 score against overlap proportion for different scenarios of logistic regression. Compared with the original logistic regression, the proposed weighted method, either with raw weights or with updated weights, achieves much better F1 scores. The proposed method with updated weights improves the results even further. More significant improvement is observed when the overlap proportion is more than 0.7. The class 1 proportion does not influence much the performance of the proposed method. The sample size 1000 gave similar results as these for sample size 5000, but with larger standard deviations for all measurements, as expected (see Table 3.1 for example).

The simulation results for classification error rate and sensitivity against overlap proportion for weighted logistic regression can be found in Figure 3.5 and Figure 3.6. The improvement of the error rate and sensitivity is very impressive with the weighted logistic regression, either with raw weights or updated weights.

The proposed weighted method also provided very good performance for SVM, KNN and classification tree (see Figure 3.2, Figure 3.3, Figure 3.4, Figure 3.7, Figure 3.8, Figure 3.9,

Figure 3.10, Figure 3.11, and Figure 3.12).

The proposed method for SVM classifier did not necessarily drop the misclassification error rate, but the increase of the sensitivity, F1 score and G score was very significant. The weighted SVM with updated weights produced similar results compared to the case with raw weights.

The weighted KNN classifier with updated weights produced slightly better results compared to the case with with raw weights, and in both cases predicted results were improved compared to the original KNN. The performance of the proposed weighted KNN was less sensitive to the overlap proportion compared to the other methods. For example, the other weighted classifiers did not provide large improvement at 50% overlap, and better results were observed when the overlap proportion was higher. In contrast the KNN classifier provided rather consistent improvement over all overlap proportions.

The weighted classification tree provided very large improvement when the overlap was high, and the updated weights introduced better result compared to the raw weights.

The robustness of the weighted model method was also tested in the simulation study (see Table 3.2). When the data was actually error-free, the weighted models were trained based on different overlap proportion assumptions. It was found that the weighted model method was quite robust to this mis-specification situation and the classification results barely changed.

3.3.2 Application on the prostate cancer image data

The proposed weighted models, i.e. the weighted logistic regression, weighted SVM, weighted KNN and weighted classification tree were applied on the prostate cancer image data. The construction of the testing set was the same as in Chapter 2.4.2. In this application we used the updated weights to fit the classifiers, and the data-adaptive function of the form (3.1) was used.

The classification results for patient 1015 are summarized in Table 3.3. The proposed weighted classifiers provide improvement on all measures and under all assumed error levels (except for weighted classification tree, for which the sensitivity is slightly decreased).

Table 3.4 shows the classification results for patient 2008. The weighted logistic regression

has no improvement, and the reason may be that the linear relationship of response and covariates does not hold. The largest improvement for the other classifiers is observed under the registration error assumption 2.33 mm, which conforms the conclusion that the registration error for patient 2008 was close to 2.33 mm. The sensitivity of weighted SVM is increased by 29% under 2.33 mm registration error, and the F1 score of weighted tree is increased by 51%.

The classification results for patient 1012 are shown in Table 3.5. It can be seen the proposed weighted classifiers improve the results under all assumed registration errors. In the cases of weighted SVM under 1.39 mm and 1.86 mm registration error, the sensitivity may not be increased or even worse, but the specificity is increased significantly. The resulting F1 score is doubled or even tripled.

The classification results for patient 1035 in Table 3.6 indicate that the weighted classifiers provide very good improvement on almost all classifiers and all assumed registration errors. The exception is weighted logistic regression under 1.39 mm and 1.86 mm registration error assumptions. In these cases, the sensitivity drops. The weighted SVM and weighted classification tree benefit most from the weighted models. For example, the weighted classification tree roughly doubles the sensitivity with registration error 1.39 mm or 1.86 mm. The F1 score for weighted SVM with registration error 2.33 mm is 2.67 times the original.

Table 3.7 presents the classification results for patient 2009. The proposed weighted models improve all measures under all assumed error levels. The improvement is the largest when the registration error is assumed 2.33 mm. Similar to the patient 1035, weighted SVM and weighted classification tree provide more improvement than weighted KNN and weighted logistic regression.

3.4 Conclusion

In this section we propose a weighted model method to eliminate the impact of misclassification in response on the model construction process. The weight is calculated and updated according

to the position of the data point. The weighted models for different classifiers are based on the idea of emphasizing more on the data with larger weights. The simulation studies indicate that the weighted model method is a very good strategy for handling misclassification in response. The updated weights usually produce better results than the raw weights scenario. The application of the weighted models on the prostate cancer image data conformed that this weighted model method could improve the classification performance comparing to directly fitting the original classifier on the error-prone data.

An important advantage of the proposed weighted model method is that it is very robust to the situation of model mis-specification. The weighted model has a limitation that it is complicated and the specific weighted model form needs to be specified before applying a classifier. In the next chapter we investigate the method that corrects the data directly so that the fitting process can be largely simplified.

3.5 Appendix

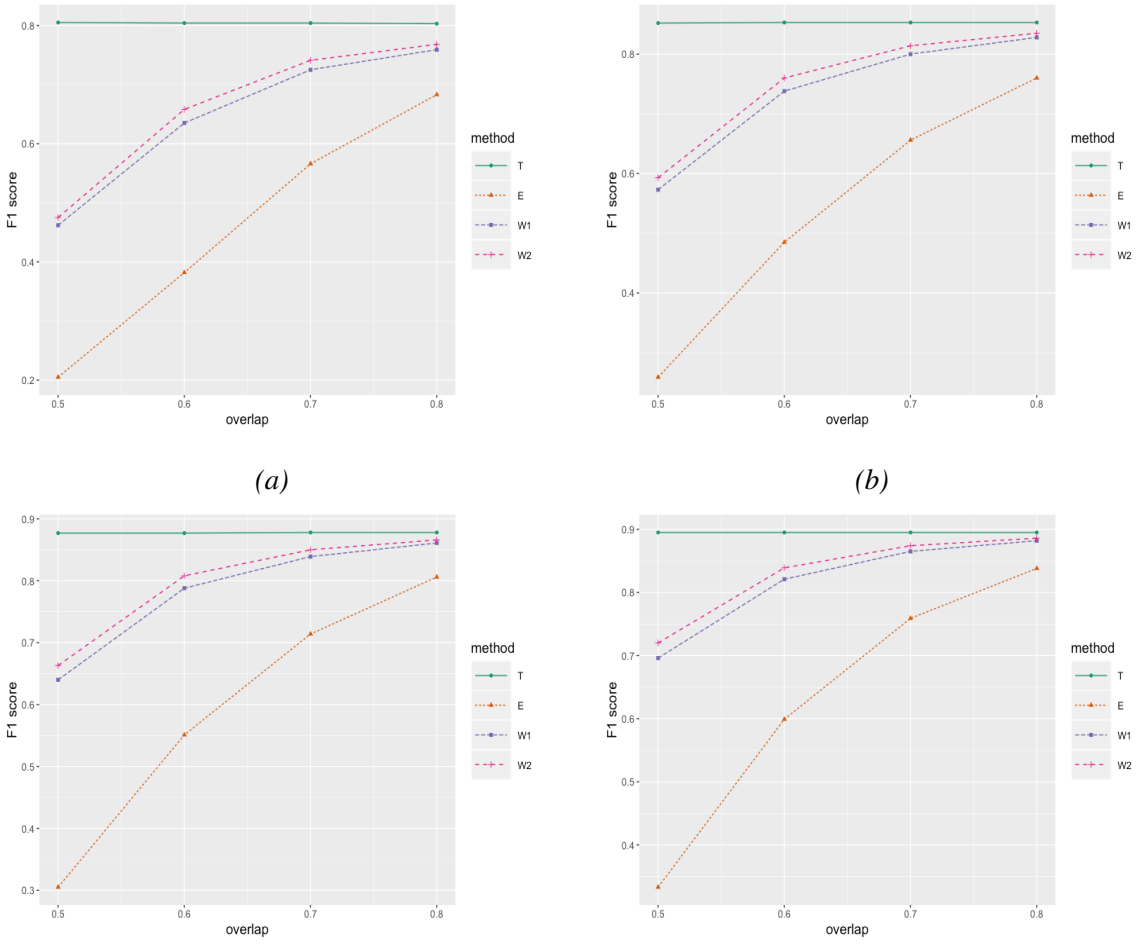


Figure 3.1: Simulated F1 score for logistic regression with different class 1 proportions. The sample size is 5000. Plot (a), (b), (c), and (d) correspond to class 1 proportion 0.05, 0.10, 0.15 and 0.20, respectively.

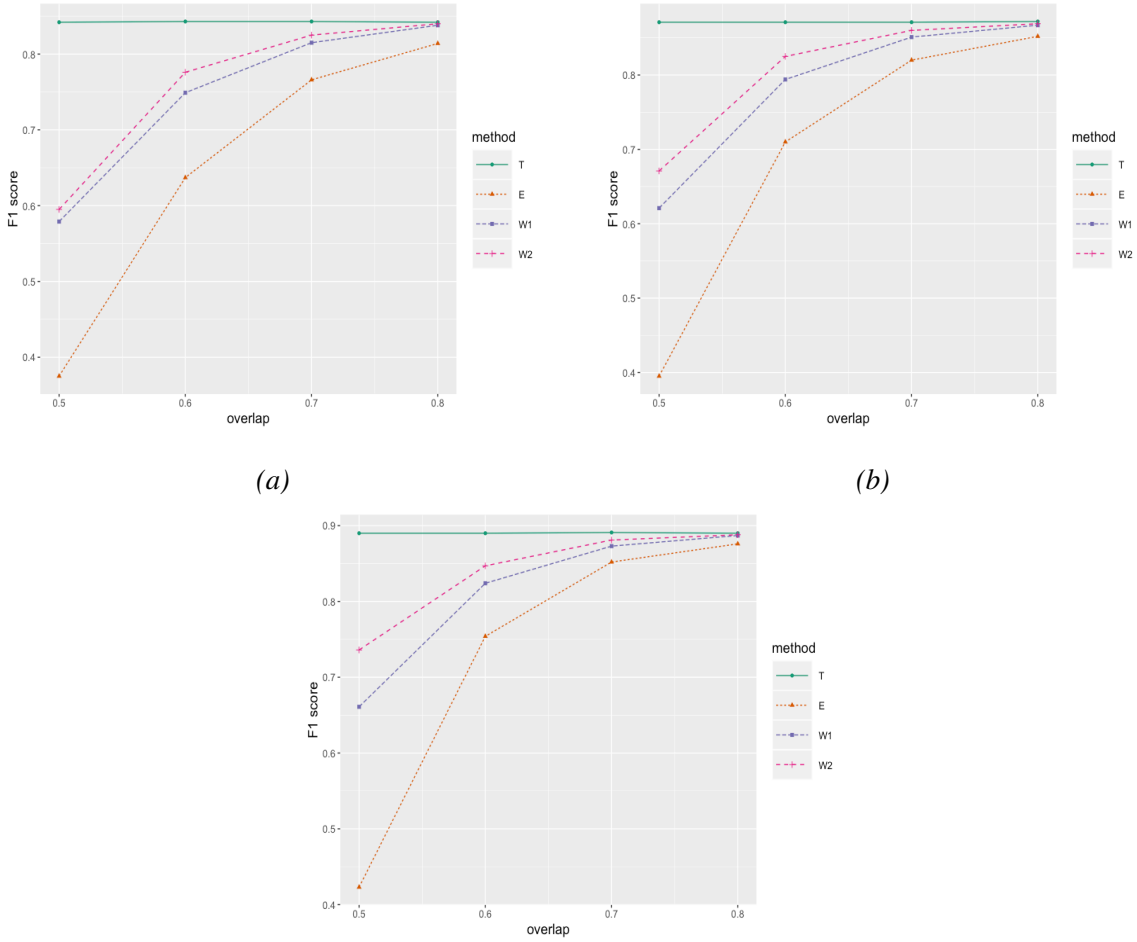
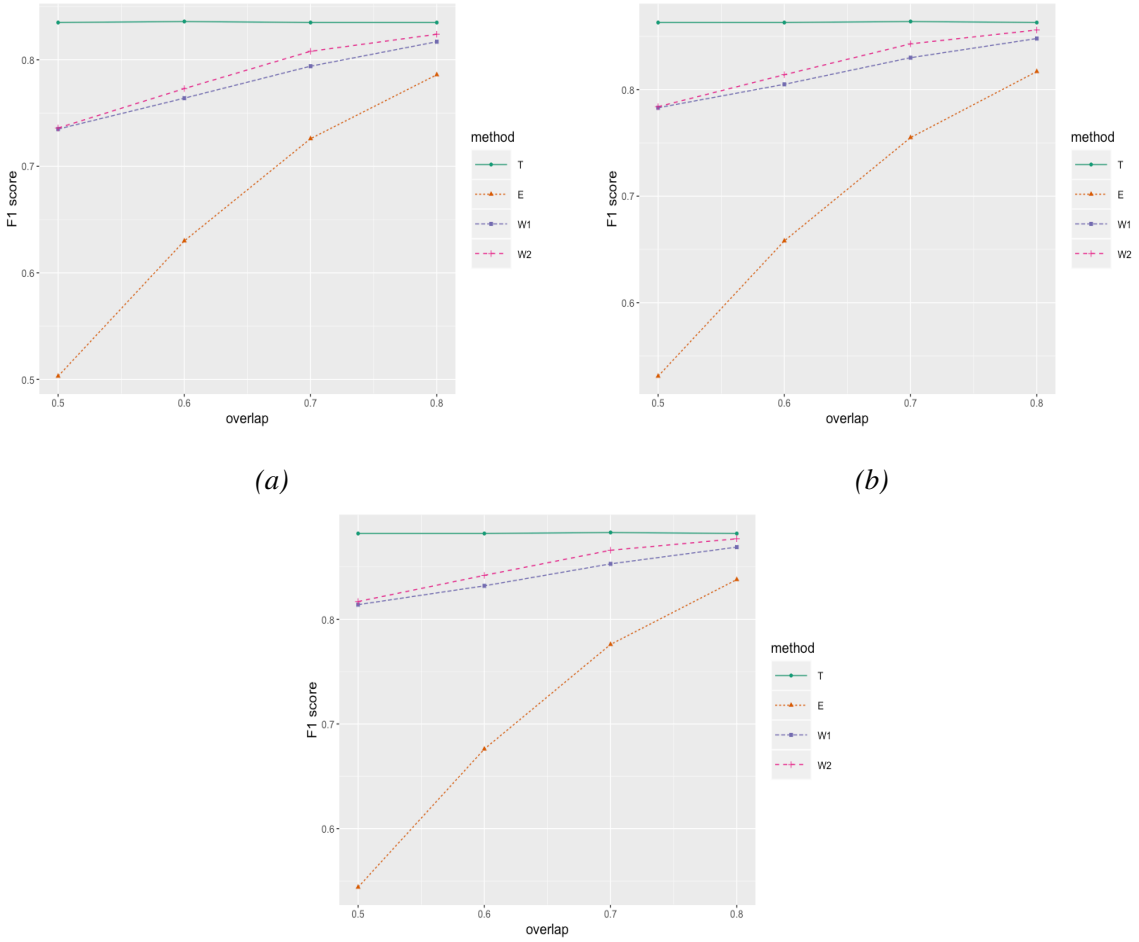


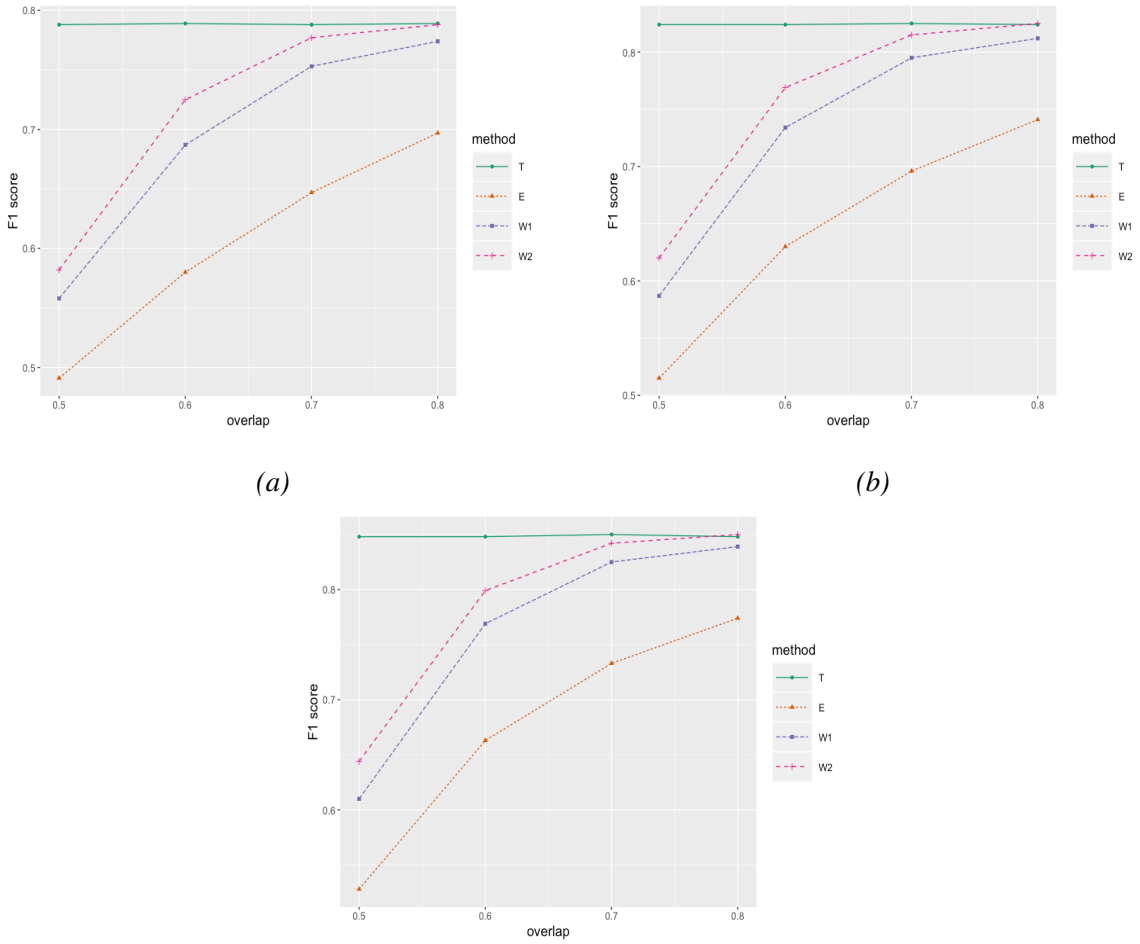
Figure 3.2: Simulated F1 score for SVM classifier with different class 1 proportions. The sample size is 5000. Plot (a), (b), and (c) correspond to class 1 proportion 0.10, 0.15 and 0.20, respectively.



(a) *(b)*

(c)

Figure 3.3: Simulated F1 score for KNN classifier with different class 1 proportions. The sample size is 5000. Plot (a), (b), and (c) correspond to class 1 proportion 0.10, 0.15 and 0.20, respectively.



(a) *(b)*

(c)

Figure 3.4: Simulated F1 score for classification tree with different class 1 proportions. The sample size is 5000. Plot (a), (b), and (c) correspond to class 1 proportion 0.10, 0.15 and 0.20, respectively.

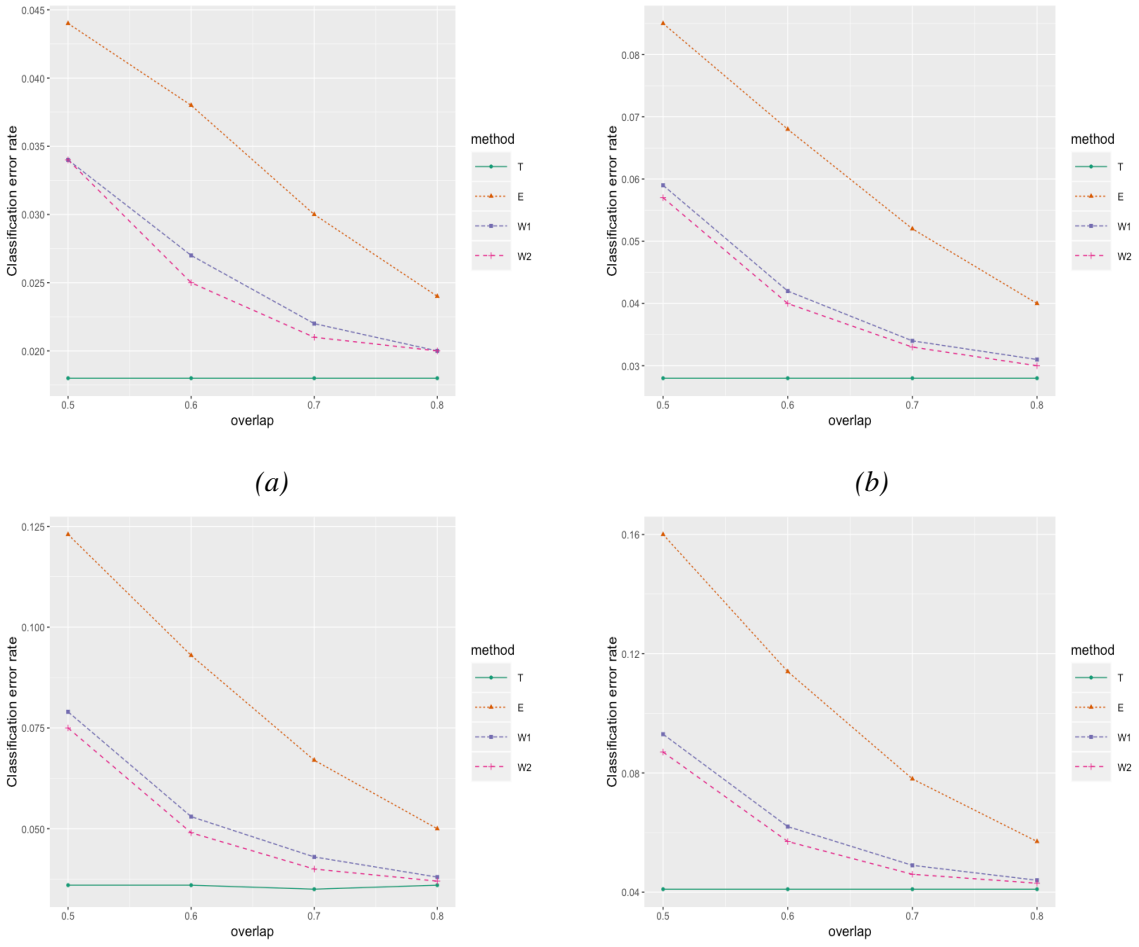


Figure 3.5: Classification error rate against overlap proportion for logistic regression with different class 1 proportions. The sample size is 5000. Plot (a), (b), (c), and (d) correspond to class 1 proportion 0.05, 0.10, 0.15 and 0.20, respectively.

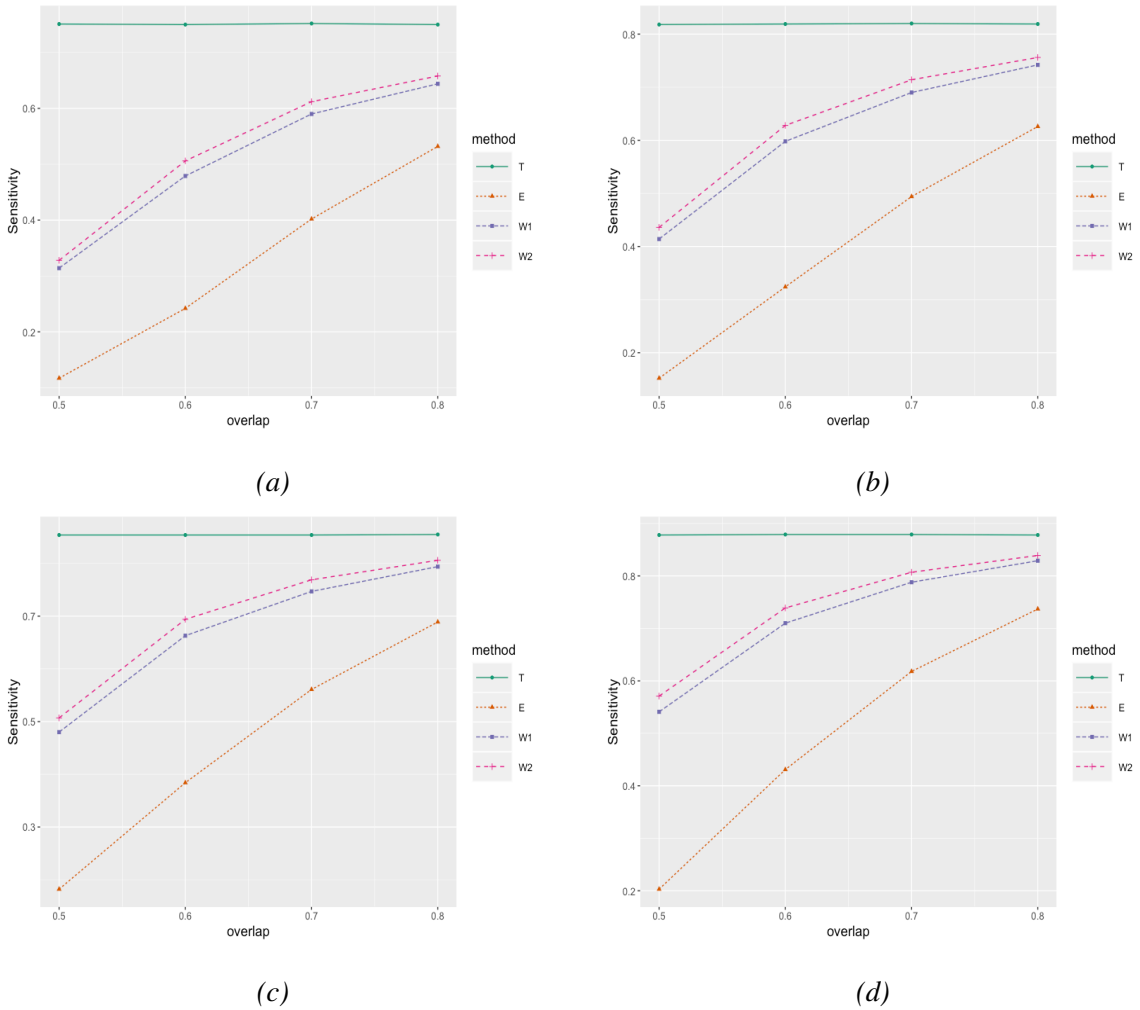


Figure 3.6: Sensitivity against overlap proportion for logistic regression with different class 1 proportions. The sample size is 5000. Plot (a), (b), (c), and (d) correspond to class 1 proportion 0.05, 0.10, 0.15 and 0.20, respectively.

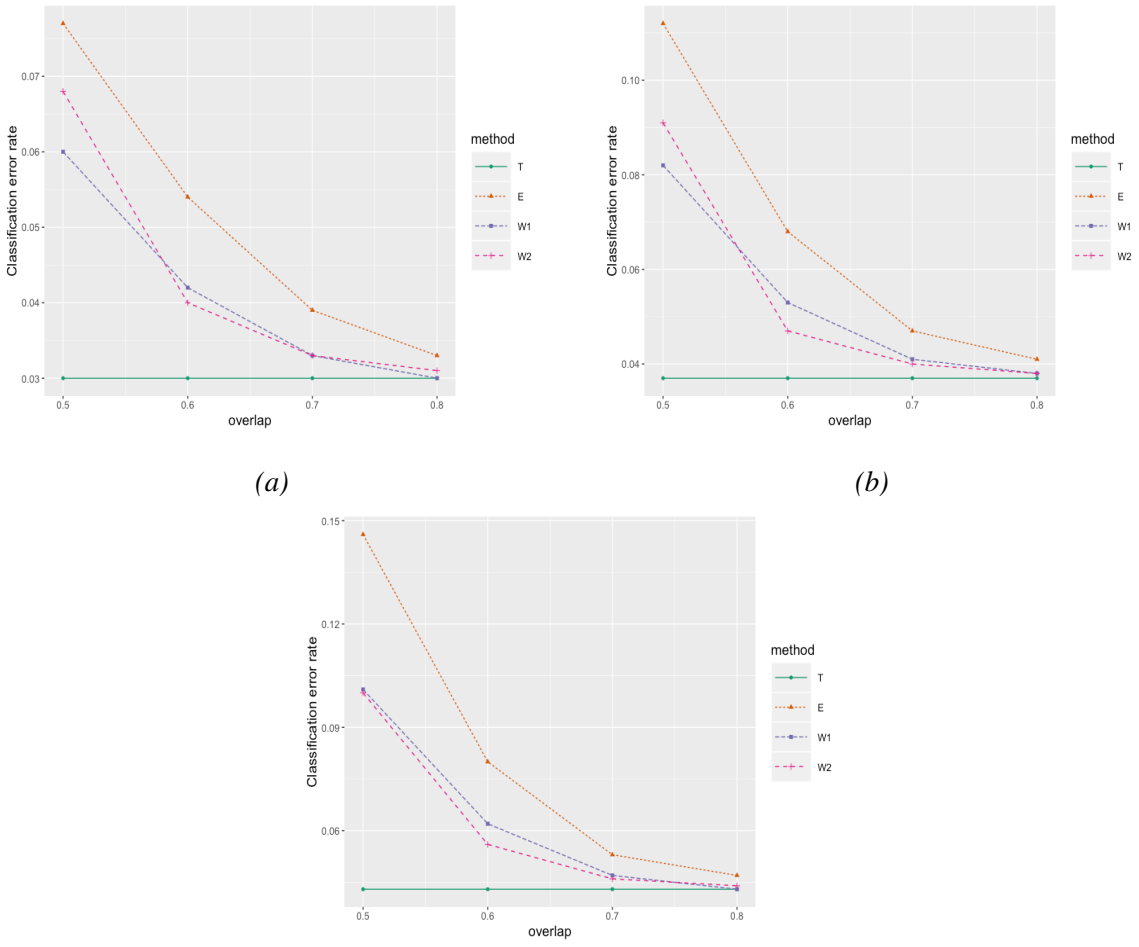


Figure 3.7: Classification error rate against overlap proportion for SVM with different class 1 proportions. The sample size is 5000. Plot (a), (b), and (c) correspond to class 1 proportion 0.10, 0.15 and 0.20, respectively.

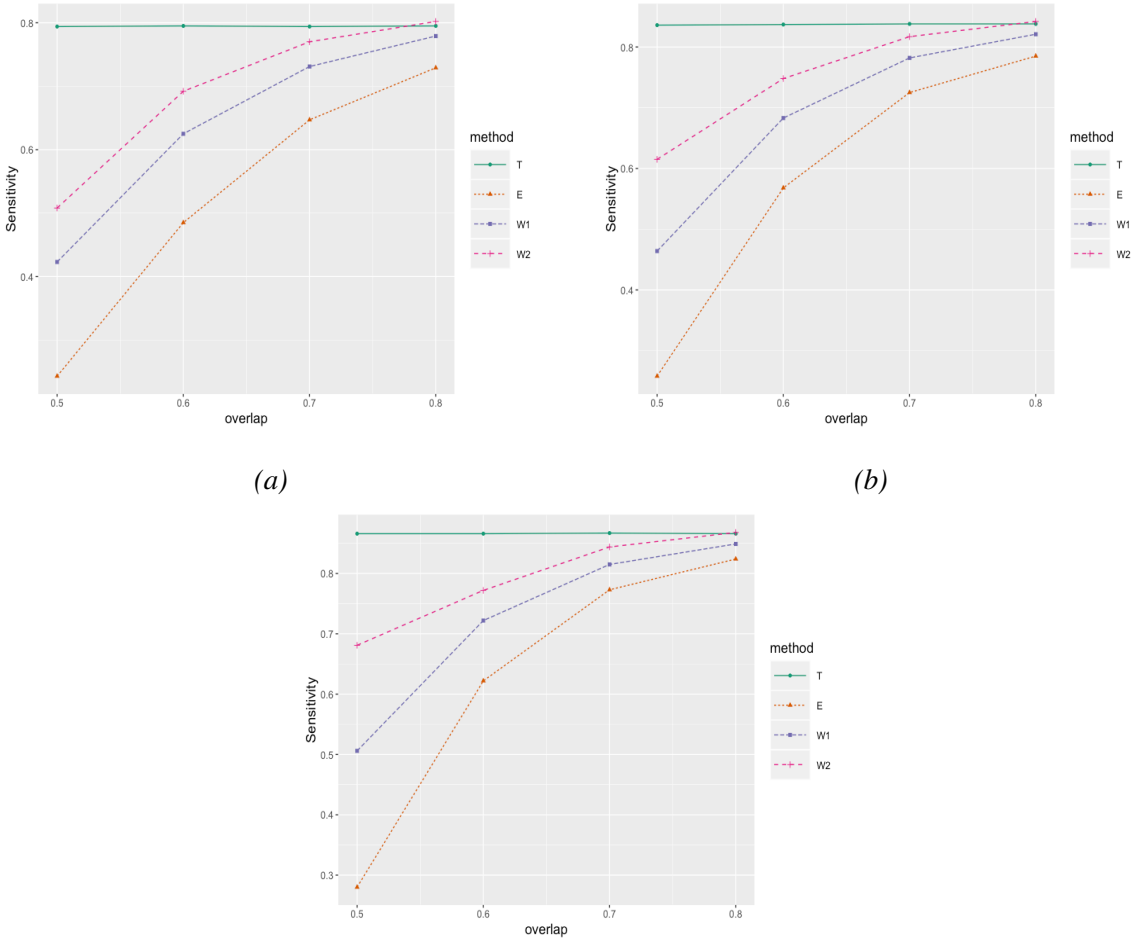


Figure 3.8: Sensitivity against overlap proportion for SVM with different class 1 proportions. The sample size is 5000. Plot (a), (b), and (c) correspond to class 1 proportion 0.10, 0.15 and 0.20, respectively.

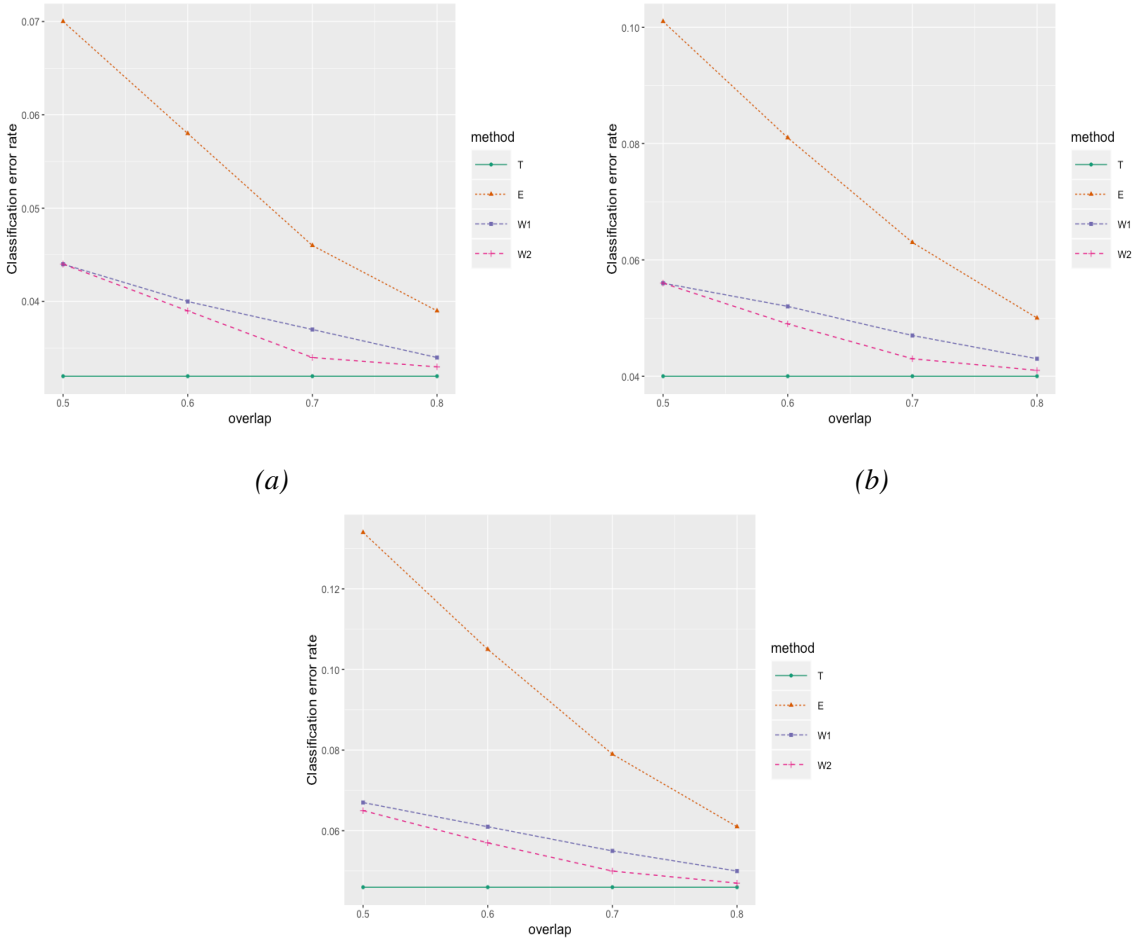
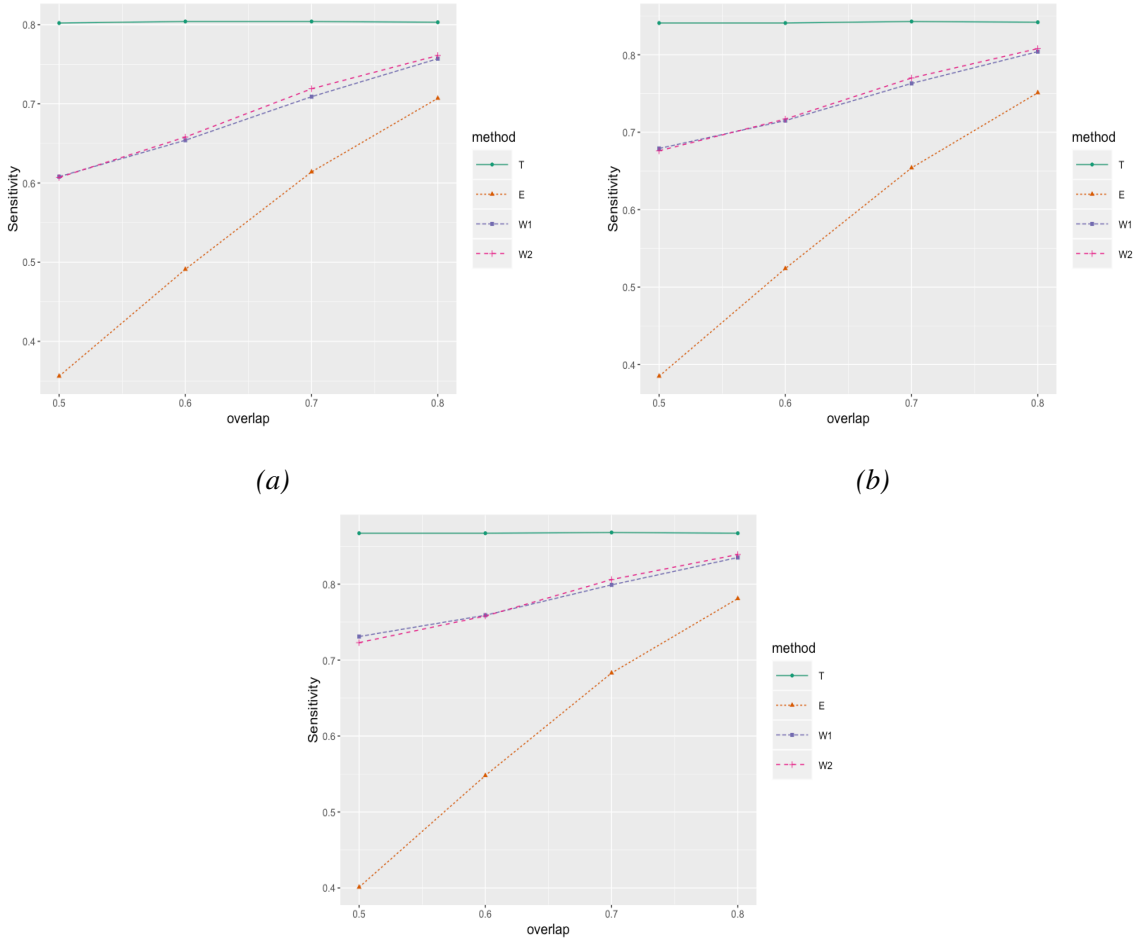


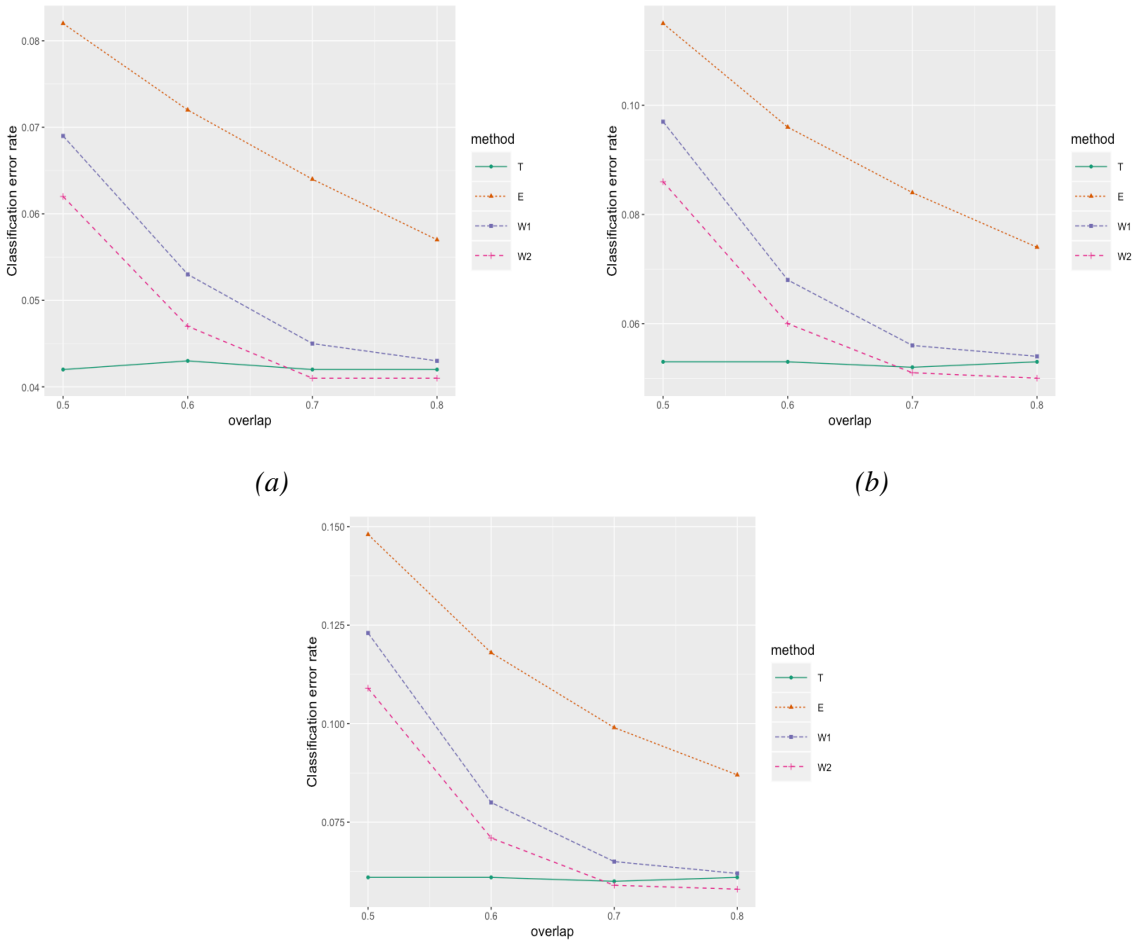
Figure 3.9: Classification error rate against overlap proportion for KNN with different class 1 proportions. The sample size is 5000. Plot (a), (b), and (c) correspond to class 1 proportion 0.10, 0.15 and 0.20, respectively.



(a) *(b)*

(c)

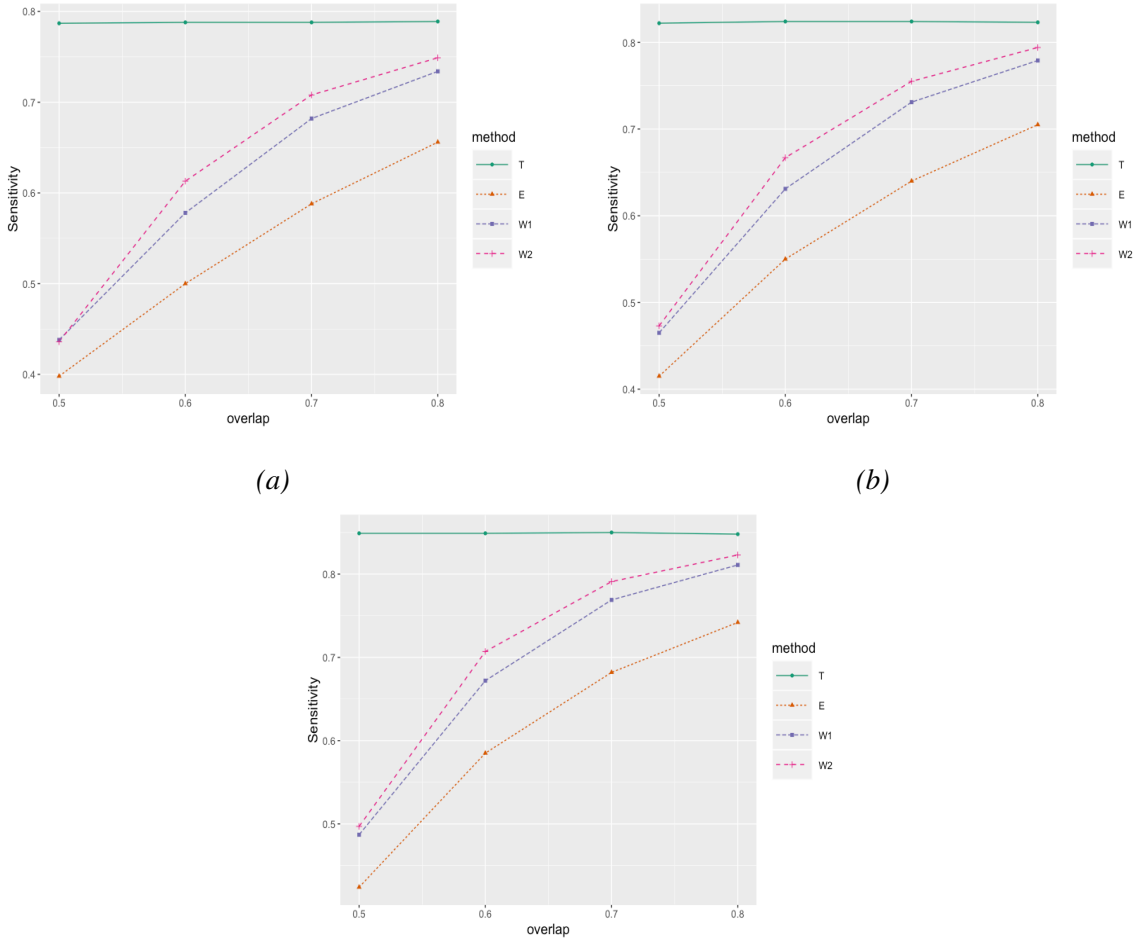
Figure 3.10: Sensitivity against overlap proportion for KNN with different class 1 proportions. The sample size is 5000. Plot (a), (b), and (c) correspond to class 1 proportion 0.10, 0.15 and 0.20, respectively.



(a) *(b)*

(c)

Figure 3.11: Classification error rate against overlap proportion for classification tree with different class 1 proportions. The sample size is 5000. Plot (a), (b), and (c) correspond to class 1 proportion 0.10, 0.15 and 0.20, respectively.



(a) *(b)*

(c)

Figure 3.12: Sensitivity against overlap proportion for Classification tree with different class 1 proportions. The sample size is 5000. Plot (a), (b), and (c) correspond to class 1 proportion 0.10, 0.15 and 0.20, respectively.

Table 3.1: The effect of sample size on the proposed method for logistic regression with class 1 proportion being 20% and overlap proportion being 70%.

sample size	methods	classification error rate	sensitivity	specificity	F1 score	G score
1000	T	0.042(0.009)	0.878(0.037)	0.979(0.008)	0.894(0.023)	0.894(0.023)
	E	0.080(0.020)	0.607(0.098)	0.998(0.003)	0.747(0.077)	0.771(0.063)
	W1	0.055(0.015)	0.754(0.078)	0.993(0.005)	0.844(0.049)	0.852(0.043)
	W2	0.051(0.014)	0.776(0.073)	0.992(0.006)	0.856(0.045)	0.862(0.040)
5000	T	0.041(0.004)	0.879(0.016)	0.979(0.004)	0.895(0.010)	0.896(0.010)
	E	0.078(0.009)	0.618(0.045)	0.998(0.001)	0.759(0.034)	0.780(0.028)
	W1	0.049(0.005)	0.788(0.027)	0.992(0.003)	0.865(0.015)	0.869(0.014)
	W2	0.046(0.005)	0.807(0.025)	0.990(0.003)	0.874(0.014)	0.877(0.013)

Table 3.2: Simulation results for the robustness of the proposed method on SVM classifier with class 1 proportion being 15% and sample size being 5000. The data is error-free and is corrected with the proposed method for different overlap proportions.

			T		
	classification error rate	sensitivity	specificity	F1	G score
	0.037(0.004)	0.837(0.025)	0.985(0.004)	0.871(0.014)	0.872(0.014)
			W1		
overlap	classification error rate	sensitivity	specificity	F1	G score
50%	0.037(0.004)	0.868(0.023)	0.980(0.005)	0.876(0.013)	0.876(0.013)
60%	0.037(0.004)	0.870(0.024)	0.980(0.005)	0.876(0.013)	0.877(0.013)
70%	0.037(0.004)	0.869(0.023)	0.980(0.005)	0.876(0.012)	0.877(0.012)
80%	0.036(0.004)	0.867(0.023)	0.981(0.005)	0.877(0.012)	0.878(0.012)
			W2		
overlap	classification error rate	sensitivity	specificity	F1	G score
50%	0.041(0.005)	0.776(0.038)	0.991(0.004)	0.849(0.021)	0.853(0.019)
60%	0.037(0.004)	0.839(0.028)	0.985(0.005)	0.871(0.014)	0.872(0.013)
70%	0.036(0.004)	0.856(0.025)	0.983(0.005)	0.876(0.013)	0.877(0.012)
80%	0.036(0.004)	0.860(0.023)	0.982(0.004)	0.878(0.012)	0.878(0.012)

Table 3.3: Classification results for patient 1015 with different classifiers.

		logistic regression		KNN		SVM		classification tree	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.076	0.058	0.100	0.095	0.137	0.103	0.116	0.098
	sensitivity	0.745	0.792	0.615	0.660	0.595	0.672	0.616	0.542
	specificity	0.946	0.961	0.935	0.935	0.895	0.925	0.917	0.946
	F1 score	0.682	0.751	0.573	0.604	0.486	0.588	0.537	0.547
1.39	classification error	0.089	0.077	0.114	0.104	0.151	0.128	0.129	0.121
	sensitivity	0.669	0.718	0.550	0.571	0.542	0.597	0.554	0.536
	specificity	0.947	0.954	0.936	0.945	0.896	0.914	0.918	0.930
	F1 score	0.663	0.709	0.558	0.591	0.486	0.551	0.529	0.537
2.33	classification error	0.065	0.052	0.089	0.072	0.129	0.100	0.107	0.103
	sensitivity	0.855	0.885	0.691	0.722	0.661	0.720	0.679	0.579
	specificity	0.944	0.955	0.934	0.949	0.893	0.918	0.916	0.929
	F1 score	0.712	0.763	0.593	0.652	0.490	0.573	0.544	0.512

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 3.4: Classification results for patient 2008 with different classifiers.

		logistic regression		KNN		SVM		classification tree	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.089	0.088	0.141	0.077	0.069	0.039	0.110	0.054
	sensitivity	0.000	0.000	0.400	0.483	0.621	0.765	0.409	0.771
	specificity	0.998	0.999	0.903	0.965	0.961	0.980	0.936	0.962
	F1 score	0.000	0.000	0.330	0.523	0.612	0.776	0.393	0.711
1.39	classification error	0.104	0.109	0.144	0.100	0.077	0.063	0.102	0.076
	sensitivity	0.000	0.000	0.398	0.411	0.550	0.700	0.390	0.581
	specificity	0.999	0.993	0.908	0.956	0.965	0.965	0.956	0.963
	F1 score	0.000	0.000	0.362	0.458	0.594	0.697	0.439	0.611
2.33	classification error	0.083	0.0816	0.141	0.071	0.065	0.033	0.109	0.075
	sensitivity	0.000	0.000	0.375	0.480	0.638	0.824	0.391	0.592
	specificity	0.998	1.000	0.902	0.968	0.961	0.979	0.936	0.954
	F1 score	0.000	0.000	0.303	0.523	0.615	0.801	0.369	0.561

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 3.5: Classification results for patient 1012 with different classifiers.

		logistic regression		KNN		SVM		classification tree	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.091	0.056	0.096	0.029	0.144	0.023	0.142	0.049
	sensitivity	0.754	0.849	0.834	0.804	0.794	0.779	0.352	0.769
	specificity	0.911	0.946	0.905	0.973	0.858	0.981	0.866	0.954
	F1 score	0.210	0.329	0.219	0.470	0.152	0.525	0.074	0.335
1.39	classification error	0.095	0.035	0.099	0.034	0.145	0.028	0.097	0.057
	sensitivity	0.691	0.303	0.763	0.747	0.763	0.684	0.230	0.743
	specificity	0.911	0.982	0.905	0.971	0.857	0.979	0.920	0.948
	F1 score	0.264	0.301	0.275	0.518	0.205	0.542	0.104	0.391
2.33	classification error	0.089	0.018	0.096	0.026	0.144	0.021	0.207	0.038
	sensitivity	0.861	0.267	0.960	0.921	0.881	0.931	0.990	0.931
	specificity	0.911	0.987	0.904	0.975	0.856	0.980	0.791	0.962
	F1 score	0.137	0.194	0.142	0.372	0.092	0.423	0.073	0.286

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 3.6: Classification results for patient 1035 with different classifiers.

		logistic regression		KNN		SVM		classification tree	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.009	0.014	0.108	0.051	0.107	0.010	0.141	0.065
	sensitivity	0.653	0.367	0.918	0.949	0.643	0.888	0.459	1.000
	specificity	0.995	0.992	0.892	0.949	0.895	0.991	0.863	0.934
	F1 score	0.637	0.364	0.163	0.300	0.120	0.667	0.069	0.261
1.39	classification error	0.009	0.010	0.098	0.061	0.100	0.029	0.124	0.046
	sensitivity	0.663	0.644	0.817	0.885	0.538	0.721	0.490	0.923
	specificity	0.995	0.994	0.903	0.939	0.905	0.974	0.881	0.954
	F1 score	0.654	0.618	0.173	0.266	0.119	0.383	0.090	0.334
2.33	classification error	0.021	0.019	0.085	0.053	0.134	0.023	0.114	0.042
	sensitivity	0.367	0.585	0.803	0.852	0.755	0.725	0.633	0.917
	specificity	0.997	0.992	0.919	0.950	0.870	0.984	0.893	0.959
	F1 score	0.499	0.634	0.347	0.472	0.240	0.640	0.237	0.548

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 3.7: Classification results for patient 2009 with different classifiers.

		logistic regression		KNN		SVM		classification tree	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.055	0.055	0.057	0.047	0.037	0.051	0.060	0.046
	sensitivity	0.000	0.015	0.747	0.791	0.769	0.834	0.615	0.726
	specificity	0.984	0.982	0.951	0.960	0.971	0.954	0.954	0.963
	F1 score	0.000	0.021	0.503	0.569	0.621	0.562	0.446	0.552
1.39	classification error	0.063	0.064	0.061	0.050	0.039	0.064	0.057	0.054
	sensitivity	0.000	0.021	0.735	0.739	0.749	0.803	0.621	0.740
	specificity	0.982	0.981	0.949	0.961	0.971	0.943	0.959	0.956
	F1 score	0.000	0.030	0.529	0.580	0.639	0.540	0.503	0.558
2.33	classification error	0.048	0.050	0.053	0.042	0.033	0.048	0.055	0.037
	sensitivity	0.000	0.006	0.770	0.814	0.787	0.845	0.617	0.841
	specificity	0.984	0.982	0.953	0.963	0.973	0.955	0.956	0.967
	F1 score	0.000	0.008	0.485	0.559	0.610	0.533	0.420	0.598

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Chapter 4

Data reconstruction method

4.1 Introduction

In the previous two chapters, we proposed to solve the misalignment problem of the prostate cancer image data in the aspect of misclassification in response. In this chapter, we consider the misalignment in the aspect of measurement error in covariates. A weighted data reconstruction method is proposed to correct the error-prone data directly. This data reconstruction method is originated from the moment reconstruction method, but combines two forms of the moment reconstruction under two assumptions. The numerical studies show the proposed method works very well in the misalignment situation.

The rest of the chapter is organized as follows. Section 4.2 describes the notations and the moment reconstruction method. In section 4.3 the details of the proposed method is presented. Simulation studies and real data application are carried out in 4.4 for different classifiers. Finally this chapter is concluded in 4.5.

4.2 Notation and framework

Let Y denote the true binary response that may not be directly observed, and the observed version of Y is Y^*. Denote $\boldsymbol{X}$ the true covariate that may not be correctly measured, and the

observed version is $\boldsymbol{X}^*$. A weight ω between 0 and 1 is assigned for each data point. The details of how to calculate the weight is discussed in section 2.3.2.

The moment reconstruction proposed by Freedman et al. (2004) introduces a moment matching method to construct an "adjusted" value for the error-prone covariate $\boldsymbol{X}^*$. The constructed variable $\boldsymbol{X}_{mr}$ is a function of $\boldsymbol{X}^*$ and Y:

$$X_{mr}(\boldsymbol{X}^*, Y) = E(\boldsymbol{X}^*|Y)(I_{p_x} - G) + \boldsymbol{X}^* G, \tag{4.1}$$

where $G = G(Y) = \{\text{cov}(\boldsymbol{X}^*|Y)^{1/2}\}^{-1}\text{cov}(\boldsymbol{X}|Y)^{1/2}$ and $A^{1/2}$ is the Cholesky decomposition of A, with I_{p_x} being the identity matrix where p_x stands for the dimension of $\boldsymbol{X}$. Under the assumption

$$E(\boldsymbol{X}^*|Y) = E(\boldsymbol{X}|Y), \tag{4.2}$$

$(\boldsymbol{X}_{mr}, Y)$ has the same first two moments as $(\boldsymbol{X}, Y)$. Freedman et al. (2004) also extended the moment reconstruction method to the situation that $\boldsymbol{X}^*$ is not an unbiased measurement of $\boldsymbol{X}$, with

$$E(\boldsymbol{X}^*|Y) = a(Y) + b(Y)E(\boldsymbol{X}|Y), \tag{4.3}$$

where $a(Y)$ and $b(Y)$ are known functions of Y. Under this assumption, the moment reconstruction becomes:

$$X^*{}_{mr}(\boldsymbol{X}^*, Y) = \frac{E(\boldsymbol{X}^*|Y) - a(Y)}{b(Y)}(I_{p_x} - G^*) + \frac{\boldsymbol{X}^* - a(Y)}{b(Y)}G^*, \tag{4.4}$$

where $G^* = G^*(Y) = b(Y)\{\text{cov}(\boldsymbol{X}^*|Y)^{1/2}\}^{-1}\text{cov}(\boldsymbol{X}|Y)^{1/2}$ and $A^{1/2}$ is the Cholesky decomposition of A, with I_{p_x} being the identity matrix where p_x is the dimension of $\boldsymbol{X}$.

4.3 The proposed method

The misalignment problem of the prostate imaging data can be viewed as the true covariates being shifted for a distance, assuming the response is correctly classified (see Figure 1.4). For example, the observed covariate value $\boldsymbol{X}_i^*$ for the point i comes from the value $\boldsymbol{X}_j$ of a nearby

point j. The weight ω_i can be viewed as a measure of how likely $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ belong to the same class:

$$\omega_i \approx \Pr(Y_i = Y_j | \boldsymbol{X}_i^* = \boldsymbol{X}_j).$$

Assuming $\boldsymbol{X}$ has different distributions for different classes, then whether $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ come from the same class has an impact on the measurement error assumption. If the mis-measured covariate $\boldsymbol{X}_j : \boldsymbol{X}_i^* = \boldsymbol{X}_j$ has the same class label as $\boldsymbol{X}_i$, then $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ have the same distribution. Thus the assumption $E(\boldsymbol{X}^*|Y) = E(\boldsymbol{X}|Y)$ is valid; if the mis-measured covariate comes from the opposite class, i.e. $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ have different class labels, then they are generated from different distributions. In this circumstance the assumption $E(\boldsymbol{X}^*|Y) = a(Y) + b(Y)E(\boldsymbol{X}|Y)$ is appropriate.

Based on the above discussion, we proposed a weighted data reconstruction method with binary response Y:

$$\tilde{X}_{mr}(\boldsymbol{X}^*, Y) = \omega X_{mr}(\boldsymbol{X}^*, Y) + (1 - \omega) X^*_{mr}(\boldsymbol{X}^*, Y), \tag{4.5}$$

where $X_{mr}(\boldsymbol{X}^*, Y)$ and $X^*_{mr}(\boldsymbol{X}^*, Y)$ are defined in equations (4.1) and (4.4).

The basic idea of the data reconstruction method is to combine the moment reconstruction from two assumptions ((4.2) and (4.3)), and the proportion assigned to each assumption is determined by the weight. For a point $(\boldsymbol{x}_i^*, y_i)$ with weight ω_i, if $y_i = 1$, the value of $\boldsymbol{x}_i^*$ comes from a point with class label 1 is ω_i, and the value of $\boldsymbol{x}_i^*$ comes from a point with class label 0 is with probability $1 - \omega_i$. The assumptions under the two situations are different. If $\boldsymbol{x}_i^*$ comes from the same class, then $E(\boldsymbol{X}^*|Y) = E(\boldsymbol{X}|Y)$ is assumed, and equation (4.1) can be applied to find an unbiased estimate for $\boldsymbol{x}_i^*$, i.e. $X_{mr}(\boldsymbol{X}^* = \boldsymbol{x}_i^*, Y = 1)$. If $\boldsymbol{x}_i^*$ comes from the different class, then it is appropriate to assume $E(\boldsymbol{X}^*|Y) = a(Y) + b(Y)E(\boldsymbol{X}|Y)$, and equation (4.4) should be used. In this case

$$\tilde{X}_{mr}(\boldsymbol{X}^* = \boldsymbol{x}_i^*, Y = 1) = \omega_i X_{mr}(\boldsymbol{X}^* = \boldsymbol{x}_i^*, Y = 1) + (1 - \omega_i) X^*_{mr}(\boldsymbol{X}^* = \boldsymbol{x}_i^*, Y = 1).$$

There is a key difference between the proposed data reconstruction method and the original moment reconstruction. There is only one error assumption in the original moment reconstruction method, while in our proposed method both assumptions are considered. This flexible

model has a drawback: the reconstructed variable $\tilde{X}_{mr}(\boldsymbol{X}^*, Y)$ is not necessarily an unbiased estimate for $\boldsymbol{X}$ given Y:

$$E(\tilde{\boldsymbol{X}}_{mr}|Y) = \omega E(\boldsymbol{X}_{mr}|Y) + (1-\omega)E(\boldsymbol{X}^*_{mr}|Y). \tag{4.6}$$

If the observed value $\boldsymbol{X}^*$ comes from the covariate in the same class of Y, then the assumption $E(\boldsymbol{X}^*|Y) = E(\boldsymbol{X}|Y)$ holds. In this case the conditional expectation (4.6) becomes

$$E(\tilde{\boldsymbol{X}}_{mr}|Y) = \omega E(\boldsymbol{X}|Y) + (1-\omega)\frac{E(\boldsymbol{X}|Y) - a(Y)}{b(Y)}.$$

The expectation $E(\tilde{\boldsymbol{X}}_{mr}|Y)$ does not equal to $E(\boldsymbol{X}|Y)$ unless $\omega = 1$. If the value of $\boldsymbol{X}^*$ comes from the covariate in the different class of Y, then the assumption $E(\boldsymbol{X}^*|Y) = a(Y) + b(Y)E(\boldsymbol{X}|Y)$ holds. The conditional expectation becomes

$$\omega\{a(Y) + b(Y)E(\boldsymbol{X}|Y)\} + (1-\omega)E(\boldsymbol{X}|Y).$$

It does not equal to $E(\boldsymbol{X}|Y)$ unless ω is 0. This indicates that the proposed constructed method produces unbiased estimates for the points far away from the cancer and non-cancer boundary. For the points near the cancer and non-cancer boundary, the proposed constructed estimates will be different.

When we predict the cancer status for a patient in the future, there is no misalignment issue since the in-vivo data is used directly without any alignment. As a result, the covariates of the future data can be viewed as error-free. Therefore, the goal is to reconstruct the training data such that the classifiers trained on the reconstructed data is close to the one trained on the error-free data. This classifier can be used for prediction purpose in the future.

When implementing the proposed method, the values of $a(Y)$ and $b(Y)$ can be estimated through the relation (4.3). The terms $\mathrm{E}(\boldsymbol{X}^*|Y)$ and $\mathrm{cov}(\boldsymbol{X}^*|Y)$ can be estimated with those training data with weight not being 1, and $\mathrm{cov}(\boldsymbol{X}|Y)$ can be estimated by those training data with weight being 1.

4.4 Numerical investigation

We present the simulation studies and real data application of the proposed method in this section. The numerical studies were done using R 3.5.2 (R Core Team, 2018). The packages *e1071* (Meyer et al., 2019), *randomForest* (Liaw et al., 2002) and *class* (Venables and Ripley, 2002) were used to perform the analysis using SVM, random forest and KNN.

4.4.1 Simulation study

We performed the simulation studies to test the performance of the proposed weighted data reconstruction method. The data generation procedure was the same as described in 2.4.1.

The detailed simulation procedure was as follows:

- step 1: simulate the true data $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, n$, $n = 1000$ or 5000;
- step 2: shift the circle of class 1 in the true data by a distance to create the error-prone data $(\boldsymbol{x}_i, y_i^*)$ according to different overlap proportions of the true class label and the observed class label. Treat the misaligned data as measurement error in covariates, i.e. $(\boldsymbol{x}_i^*, y_i)$;
- step 3: calculate the raw weights with equation (2.4) and reconstruct the data with the proposed method with the raw weights;
- step 4: update the raw weights with method described in Chapter 2.3.2. The model used to estimate the new weight (probability) is logistic regression. Reconstruct the data with the updated weights.

To compare the proposed method and the original moment reconstruction method, the following scenarios were considered in the simulation:

- T: the classifier is trained and tested on the error-free data set $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, n$.
- E: the classifier is trained and tested on the error-corrupted data set $(\boldsymbol{x}_i^*, y_i)$, $i = 1, \ldots, n$.

- MR0: the original moment reconstruction (4.4) is used to reconstruct the data set, and the classifier is trained on the reconstructed data set.

- MR1: the proposed moment reconstruction with raw weights is used for reconstruction, and the classifier is trained on the reconstructed data set.

- MR2: the proposed moment reconstruction with updated weights is used for reconstruction, and the classifier is trained on the reconstructed data set.

The data was randomly split into half training and half testing data, so in each fitting process the training set was reconstructed and used to train the model, and the testing set was used to test the performance by comparing the predicted class labels to the true class labels. The overlap proportion of the true response and the observed response ranged from 0.5 to 0.8. The class 1 proportion ϕ took the value of 0.1, 0.15, and 0.2. Each scenario was repeated 1000 times. The classification error rate, sensitivity, specificity, F1 score, and G score were recorded.

Logistic regression, SVM, KNN, and random forest classifier were considered in the simulation. The kernel used for SVM was radius basis, with gamma being 0.5, and penalty parameter being 100. The number of nearest neighbors considered for KNN was 5. The number of trees built for random forest classifier was 500, and in each split only one covariate was considered.

In the simulation studies there were two parameters need to be estimated for the proposed model: $a(Y)$ and $b(Y)$. To simplify the estimation procedure, we set $b(Y)$ to be a vector of $\mathbf{1}$ for both values of Y, and $a(Y)$ was estimated by $E(\boldsymbol{X}^*|Y) - E(\boldsymbol{X}|Y)$, where $E(\boldsymbol{X}^*|Y)$ was estimated by the mean value of the points with weight less than 1, and $E(\boldsymbol{X}|Y)$ was estimated as the mean value of the points with weight being 1.

Figure 4.1 shows the simulation results for KNN classifier with class 1 proportion ϕ being 0.15, and sample size being 5000. It can be seen that the estimation of the weight for each point has a huge impact on the correction for the proposed method. The method MR1 is the proposed method with the raw weights, which shows some improvement compared to the scenario without correction. The improvement is smaller than the method MR0, which is the

original moment reconstruction method proposed by Freedman et al. (2004). With updated weights, the proposed method MR2 shows significant improvement in all measures and the improvement is much larger than that from MR0 or MR1. Similar results were observed for SVM and random forest classifier (see Figure 4.2 and 4.3). The improvement brought by the proposed method with updated weights is more obvious for smaller overlap proportions, and the improvement is consistent for all class 1 proportions.

The proposed method with updated weights (MR2) performs less effectively for logistic regression. In Figure 4.4 the performance of MR2 is only slightly better than the original moment reconstruction method MR0. Compared to the other classifiers, it can be seen that MR2 does not outperform MR0 not because MR2 does not perform well, but MR0 is really effective with logistic regression.

The simulation study was also carried out to test the robustness of the weighted data reconstruction method (Table 4.1). It was found when the data was error-free, but was reconstructed under different error-levels, the performance of the classifiers trained on the reconstructed data set suffered. This is due to the fact that the reconstructed data is not unbiased of the true data, so reconstruct the error-free data may introduce a biased covariate error to the data set.

4.4.2 Application on the prostate cancer image data

The proposed weighted data reconstruction method was applied on the prostate cancer image data. The detailed data processing and testing set constructing procedure can be found in section 2.4.2. The updated weights were used to reconstruct the data set.

Table 4.2 summarizes the classification results for patient 1015. It can be seen the proposed data reconstruction method improves the classification results for all classifiers under all error levels. The improvement does not differ too much among three registration errors.

In Table 4.3, the classification results for patient 2008 are presented. All classification results get improved with the proposed method except for logistic regression. It can be found that the greatest improvement is achieved when the registration error is 2.33 mm, which is

consistent with the findings in Chapter 2 and Chapter 3. The sensitivity is doubled with KNN when the registration error is assumed 2.33 mm compared to the result on the original data, and the sensitivity and F1 score with SVM are more than doubled.

For patient 1012, the classification results are summarized in Table 4.4. The table indicates that almost all classification results are improved with the reconstructed data, especially for logistic regression. For example, when the registration error is 2.33 mm, the sensitivity of logistic regression with the reconstructed data is improved by 16%, and the F1 score is doubled.

Table 4.5 presents the results for patient 1035 with the proposed method. The reconstructed data improves the classification results significantly for all classifiers with all three error levels.

Similar to patient 1035, the proposed method produces significant improvement for all classifiers for patient 2009, as shown in Table 4.6. Particularly, the sensitivity for logistic regression increases from 0 to more than 0.45 for all three error levels.

Compared to the previous two correction methods, the data reconstruction method provided the most significant improvement, especially in sensitivity.

4.5 Conclusion

In this chapter we propose a weighted data reconstruction method to eliminate the effect of misalignment problem. The response is treated error-free while the true value of the covariate is not observed. The proposed method is based on the original moment reconstruction method proposed by Freedman et al. (2004), but considers different assumptions. The weight for each data point is used to combine the two assumptions and make the reconstructed data set a weighted version for both cases. The simulation studies show the proposed method works very well on all the classifiers. The proposed method depends on the correct estimation of the weights, once the estimation of the weights is relatively good, it outperforms the original moment reconstruction method.

The application of the proposed method on the prostate cancer image data also shows

significant improvement compared to directly using the original data.

The weighted data reconstruction method has a simple implementation. There is no need to change the forms of the classifiers, and only the training set is reconstructed. The shortcoming of this method is that it is not robust when the data is actually error-free, but is reconstructed with a registration error assumption.

4.6 Appendix

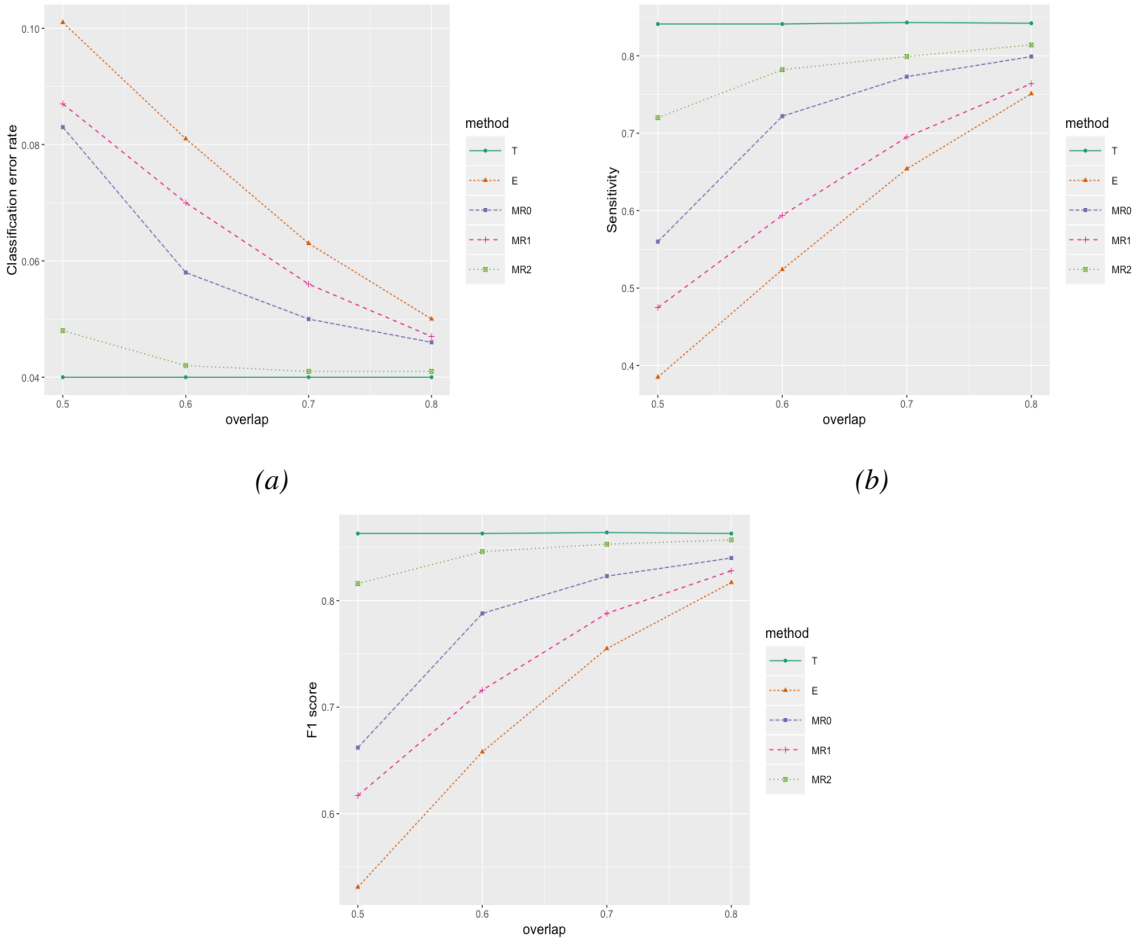


Figure 4.1: Classification error rate, sensitivity, and F1 score against overlap proportion for KNN classifier with class 1 proportion ϕ being 0.15 and sample size being 5000.

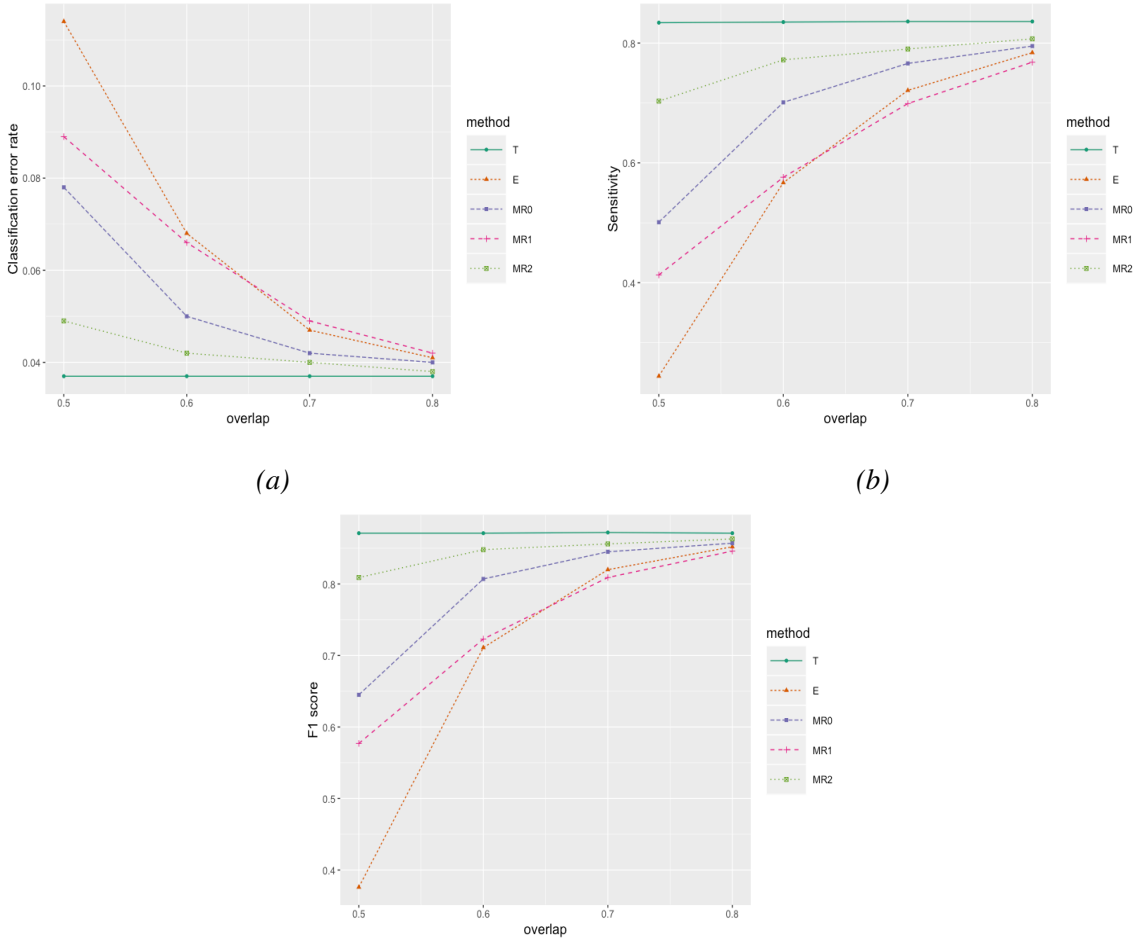


Figure 4.2: Classification error rate, sensitivity, and F1 score against overlap proportion for SVM classifier with class 1 proportion ϕ being 0.15 and sample size being 5000.

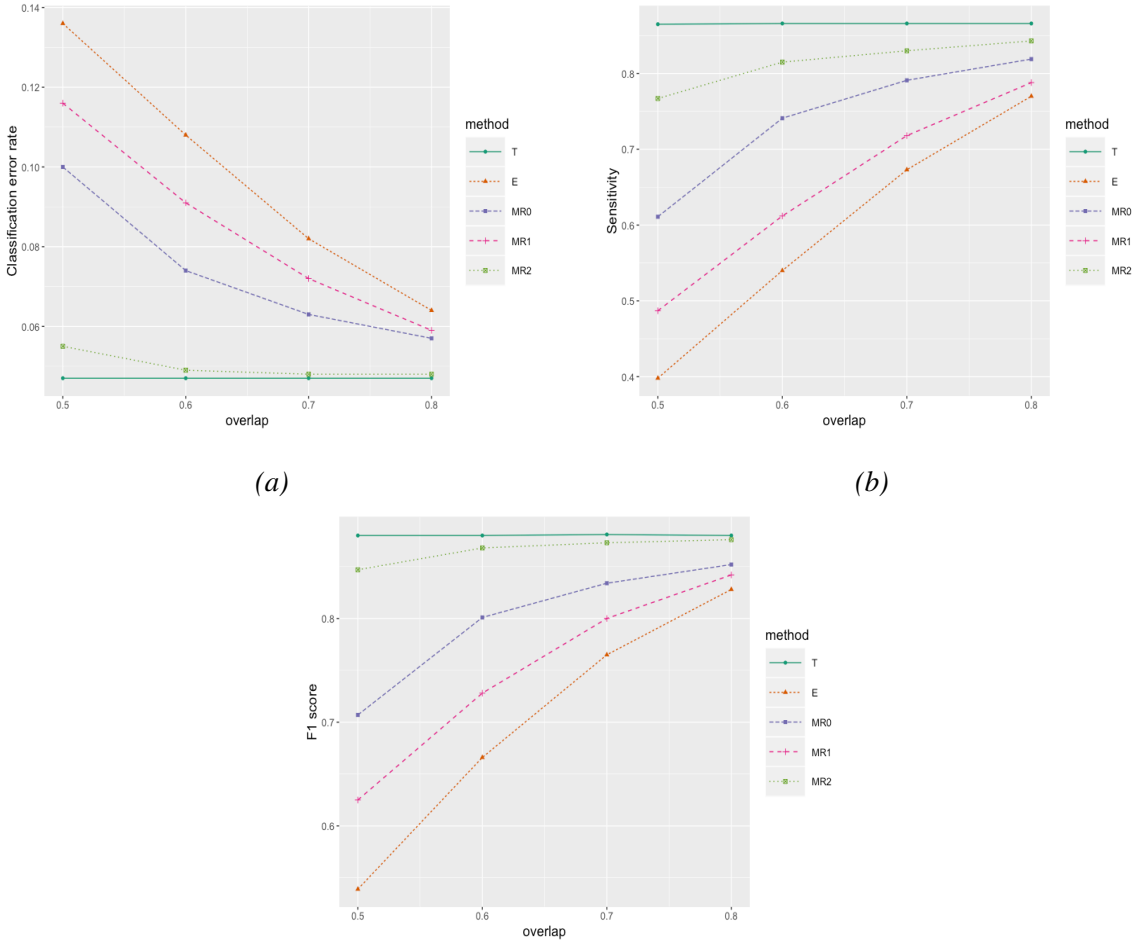


Figure 4.3: Classification error rate, sensitivity, and F1 score against overlap proportion for random forest classifier with class 1 proportion ϕ being 0.20 and sample size being 5000.

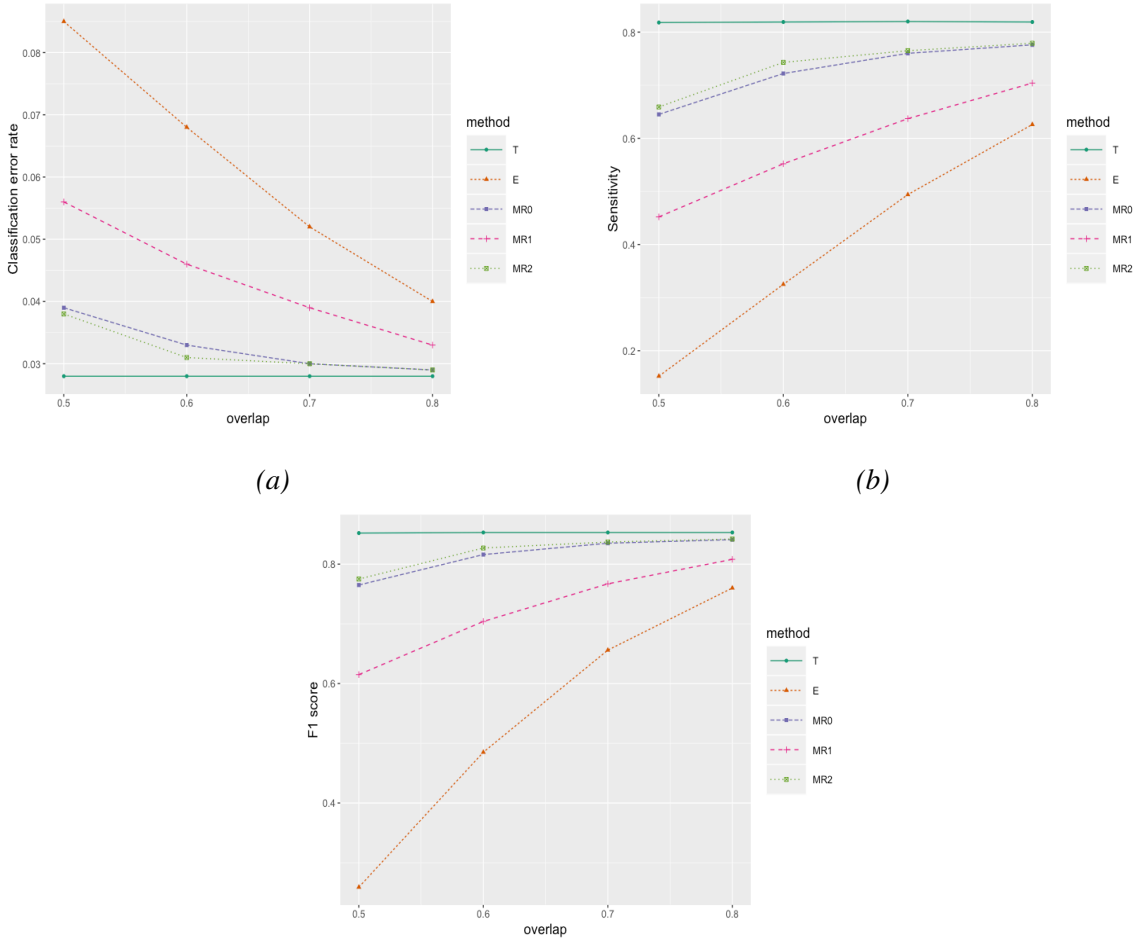


Figure 4.4: Classification error rate, sensitivity, and F1 score against overlap proportion for logistic regression with class 1 proportion ϕ being 0.10 and sample size being 5000.

Table 4.1: Simulation results for the robustness of the proposed method on KNN classifier with class 1 proportion being 10% and sample size being 5000. The data is error-free and is corrected with the proposed method for different overlap proportions.

			T		
	classification error rate	sensitivity	specificity	F1	G score
	0.032(0.004)	0.803(0.030)	0.987(0.003)	0.835(0.019)	0.836(0.019)
			MR2		
overlap	classification error rate	sensitivity	specificity	F1	G score
50%	0.034(0.005)	0.725(0.045)	0.993(0.003)	0.810(0.029)	0.817(0.026)
60%	0.032(0.004)	0.748(0.035)	0.992(0.002)	0.823(0.022)	0.827(0.021)
70%	0.032(0.004)	0.765(0.035)	0.991(0.003)	0.828(0.021)	0.831(0.020)
80%	0.032(0.004)	0.779(0.034)	0.989(0.003)	0.831(0.020)	0.834(0.019)
			MR3		
overlap	classification error rate	sensitivity	specificity	F1	G score
50%	0.050(0.008)	0.512(0.078)	0.999(0.001)	0.669(0.067)	0.706(0.053)
60%	0.041(0.005)	0.618(0.053)	0.997(0.001)	0.751(0.039)	0.771(0.032)
70%	0.035(0.004)	0.696(0.042)	0.995(0.002)	0.798(0.027)	0.808(0.024)
80%	0.033(0.004)	0.747(0.037)	0.992(0.003)	0.820(0.023)	0.825(0.021)

Table 4.2: Classification results for patient 1015 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.076	0.078	0.100	0.093	0.137	0.105	0.083	0.085
	sensitivity	0.745	0.816	0.615	0.649	0.595	0.697	0.565	0.631
	specificity	0.946	0.935	0.935	0.939	0.895	0.919	0.961	0.950
	F1 score	0.682	0.697	0.573	0.604	0.486	0.592	0.599	0.619
1.39	classification error	0.089	0.090	0.114	0.108	0.151	0.131	0.102	0.109
	sensitivity	0.669	0.731	0.550	0.575	0.542	0.588	0.480	0.518
	specificity	0.947	0.937	0.936	0.940	0.896	0.912	0.961	0.947
	F1 score	0.663	0.681	0.558	0.583	0.486	0.542	0.552	0.554
2.33	classification error	0.065	0.062	0.089	0.078	0.129	0.090	0.071	0.066
	sensitivity	0.855	0.923	0.691	0.721	0.661	0.749	0.638	0.716
	specificity	0.944	0.940	0.934	0.943	0.893	0.927	0.959	0.957
	F1 score	0.712	0.737	0.593	0.635	0.490	0.609	0.627	0.671

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 4.3: Classification results for patient 2008 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.089	0.156	0.141	0.097	0.069	0.072	0.116	0.086
	sensitivity	0.000	0.002	0.400	0.719	0.621	0.743	0.304	0.714
	specificity	0.998	0.924	0.903	0.921	0.961	0.945	0.940	0.933
	F1 score	0.000	0.003	0.330	0.563	0.612	0.642	0.314	0.590
1.39	classification error	0.104	0.175	0.144	0.130	0.077	0.077	0.123	0.121
	sensitivity	0.000	0.000	0.398	0.413	0.550	0.590	0.301	0.365
	specificity	0.999	0.919	0.908	0.923	0.965	0.961	0.942	0.937
	F1 score	0.000	0.000	0.362	0.395	0.594	0.612	0.334	0.382
2.33	classification error	0.083	0.148	0.141	0.090	0.065	0.062	0.116	0.084
	sensitivity	0.000	0.010	0.375	0.756	0.638	0.743	0.278	0.713
	specificity	0.998	0.926	0.902	0.924	0.961	0.956	0.938	0.934
	F1 score	0.000	0.011	0.303	0.579	0.615	0.663	0.281	0.579

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 4.4: Classification results for patient 1012 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.091	0.053	0.096	0.038	0.144	0.032	0.054	0.032
	sensitivity	0.754	0.844	0.834	0.834	0.794	0.799	0.794	0.869
	specificity	0.911	0.948	0.905	0.964	0.858	0.971	0.948	0.970
	F1 score	0.210	0.339	0.219	0.416	0.152	0.450	0.321	0.469
1.39	classification error	0.095	0.074	0.099	0.040	0.145	0.039	0.057	0.040
	sensitivity	0.691	0.776	0.763	0.757	0.763	0.720	0.766	0.796
	specificity	0.911	0.929	0.905	0.965	0.857	0.968	0.947	0.964
	F1 score	0.264	0.339	0.275	0.481	0.205	0.479	0.399	0.493
2.33	classification error	0.089	0.041	0.096	0.030	0.144	0.029	0.054	0.027
	sensitivity	0.861	1.000	0.960	0.960	0.881	0.960	0.960	0.980
	specificity	0.911	0.959	0.904	0.970	0.856	0.971	0.946	0.973
	F1 score	0.137	0.287	0.142	0.342	0.092	0.350	0.228	0.376

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 4.5: Classification results for patient 1035 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.009	0.034	0.108	0.038	0.107	0.019	0.060	0.023
	sensitivity	0.653	0.786	0.918	1.000	0.643	0.969	0.643	0.918
	specificity	0.995	0.968	0.892	0.962	0.895	0.981	0.943	0.977
	F1 score	0.637	0.346	0.163	0.377	0.120	0.538	0.196	0.474
1.39	classification error	0.009	0.033	0.098	0.057	0.100	0.042	0.057	0.039
	sensitivity	0.663	0.817	0.817	0.885	0.538	0.702	0.519	0.885
	specificity	0.995	0.969	0.903	0.942	0.905	0.960	0.948	0.962
	F1 score	0.654	0.386	0.173	0.275	0.119	0.292	0.184	0.363
2.33	classification error	0.021	0.025	0.085	0.059	0.134	0.037	0.052	0.039
	sensitivity	0.367	0.721	0.803	0.852	0.755	0.751	0.537	0.799
	specificity	0.997	0.982	0.919	0.944	0.870	0.969	0.959	0.966
	F1 score	0.499	0.613	0.347	0.447	0.240	0.532	0.364	0.533

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Table 4.6: Classification results for patient 2009 with different classifiers.

		logistic regression		KNN		SVM		random forest	
d [a]		R_0 [b]	R_1 [c]	R_0	R_1	R_0	R_1	R_0	R_1
1.86	classification error	0.055	0.049	0.057	0.047	0.037	0.041	0.045	0.036
	sensitivity	0.000	0.456	0.747	0.897	0.769	0.858	0.675	0.891
	specificity	0.984	0.970	0.951	0.955	0.971	0.963	0.966	0.967
	F1 score	0.000	0.417	0.503	0.598	0.621	0.619	0.537	0.661
1.39	classification error	0.063	0.057	0.061	0.053	0.039	0.043	0.047	0.042
	sensitivity	0.000	0.461	0.735	0.876	0.749	0.894	0.678	0.860
	specificity	0.982	0.967	0.949	0.950	0.971	0.960	0.966	0.963
	F1 score	0.000	0.431	0.529	0.605	0.639	0.656	0.571	0.655
2.33	classification error	0.048	0.047	0.053	0.044	0.033	0.037	0.041	0.033
	sensitivity	0.000	0.453	0.770	0.870	0.787	0.849	0.696	0.892
	specificity	0.984	0.970	0.953	0.959	0.973	0.967	0.968	0.969
	F1 score	0.000	0.387	0.485	0.565	0.610	0.599	0.526	0.635

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: proposed method

Chapter 5

Conclusion and future work

5.1 Conclusions and discussions

This thesis was motivated by the prostate cancer imaging study performed in the University of Western Ontario. In the study the in-vivo and histology images of all prostate cancer patients were taken, and the goal was to build a relationship between the in-vivo measurements and the cancer status on the histology.

In the study the in-vivo image was aligned to the histology image with certain registration procedures. However, the mapping of the two images was not perfect, and registration error was introduced in the alignment process. The registration error was caused by the shift of the two images, and could be viewed either as misclassification in response or as measurement error in covariates. The simulation studies showed the registration error may cause a large decrease in the classification performance for different classifiers.

The objective of the research was then to build classifiers to classify cancer status on histology based on the in-vivo measurements, and at the same time eliminate the impact of registration error on the classification performance.

Three methods were discussed to achieve this objective. First, the predict probability correction method based on the relationship of the probability of the observed class label and

the probability of the true class label was proposed. This method corrects the classification probability of the observed class label so that the corrected result is close to the classification probability of the true class label. The predicted class probability of the training set is corrected so that a model can be built with the estimated true class label probability and the covariates. With this model the true class probability of a new instance can be predicted.

Second, we proposed to incorporate the weight of each data point in the model construction. The weight is calculated with the position information of each data point, and it represents the reliability of each instance. Weighted logistic regression, weighted SVM, weighted KNN and weighted classification tree were introduced in Chapter 3. These weighted models can be directly used for future classification.

Lastly, the weight was incorporated in the weighted data reconstruction method to combine the different forms of moment reconstruction under two assumptions. The training set is reconstructed using the proposed weighted data reconstruction method, and the reconstructed set can be used to train different classifiers.

The above three proposed methods deal with the registration error differently. The predict probability correction method takes the registration error as misclassification in response, and works with the predictions. The weighted model method also treats the registration error as misclassification in response, but modifies the classifiers so that the misclassification in response is embedded in the model construction. The weighted data reconstruction method treats the registration error differently as the measurement error in covariates, and it creates an "adjusted" value of the error-corrupted covariates for each instance in the training set before any attempts of model fitting.

The three methods have different advantages and shortcomings. The predict probability correction method and the weighted data reconstruction method are relatively simple to implement, but suffer in the lack of robustness. The weighted models are quite robust, but is more complicated compared to the other two.

All the three proposed methods showed significant improvement in the classification per-

Table 5.1: The comparison of the three proposed methods on patient 2008 with KNN classifier.

d[a]		R_0[b]	R_1[c]	R_2[d]	R_3[e]
1.86	classification error	0.141	0.063	0.077	0.097
	sensitivity	0.400	0.432	0.483	0.719
	specificity	0.903	0.985	0.965	0.921
	F1 score	0.330	0.545	0.523	0.563
	G score	0.335	0.565	0.524	0.577
1.39	classification error	0.144	0.088	0.100	0.130
	sensitivity	0.398	0.378	0.411	0.413
	specificity	0.908	0.974	0.956	0.923
	F1 score	0.362	0.469	0.458	0.395
	G score	0.363	0.484	0.461	0.396
2.33	classification error	0.141	0.055	0.071	0.090
	sensitivity	0.375	0.473	0.480	0.756
	specificity	0.902	0.986	0.968	0.924
	F1 score	0.303	0.582	0.523	0.579
	G score	0.308	0.598	0.525	0.595

[a] d: registration error
[b] R_0: no correction situation
[c] R_1: predict probability correction method
[d] R_2: weighted model method
[e] R_3: data reconstruction method

formance in both simulation studies and real data application.

The Figure 5.1 compares the three proposed correction methods on logistic regression in the simulation study. T is the classification results on the error-free data, and E is the results for the error-prone data without any correction. C2, W2 and MR2 show the classification performance for prediction correction method, weighted model and data reconstruction method with updated weights, respectively. All three methods improve the classification performance compared to directly fitting the logistic regression on the error-prone data. The data reconstruction method shows the most significant improvement, and the predict probability correction method improves the least.

Table 5.1 summarizes the classification results for K-nearest neighbors on patient 2008 with

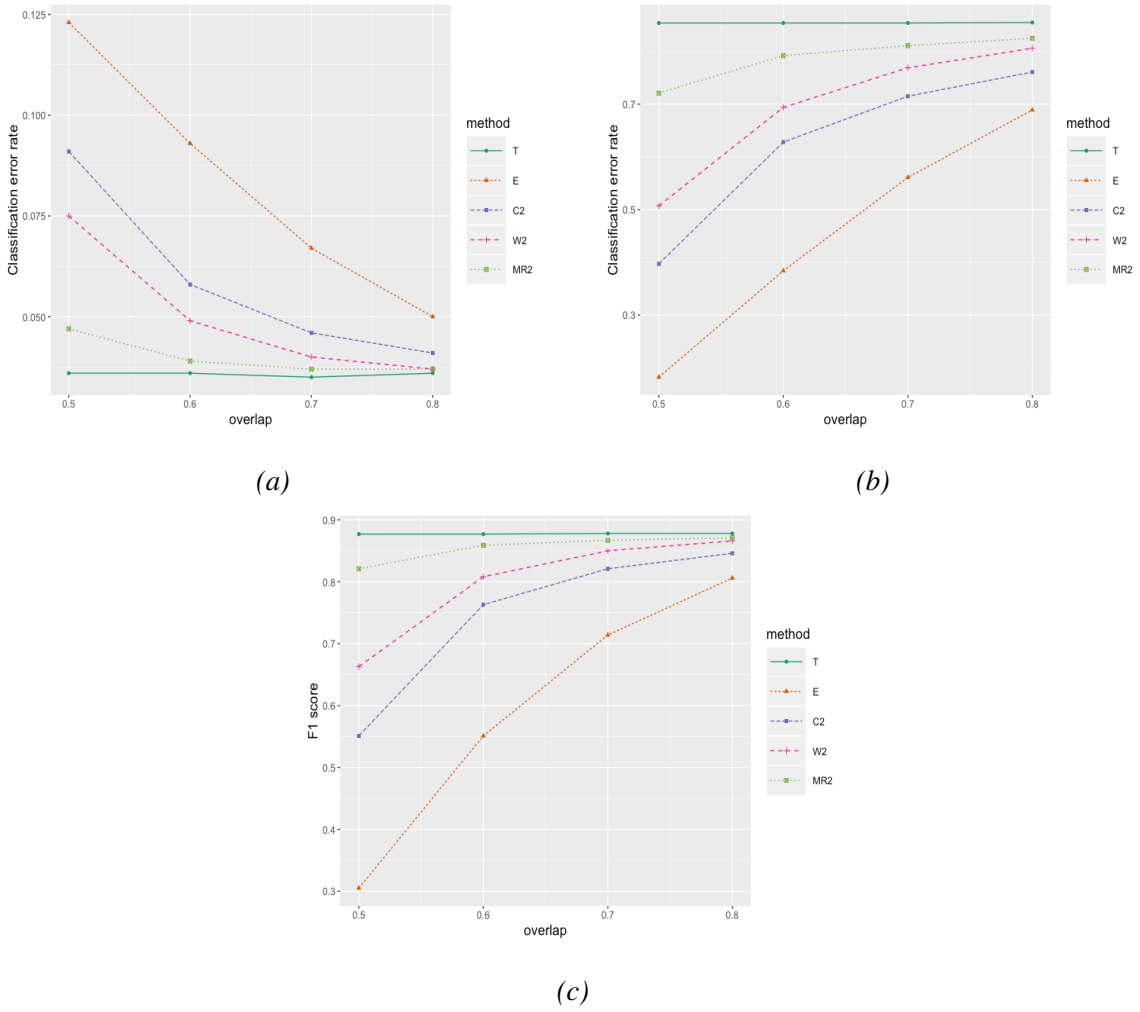


Figure 5.1: The comparison of the three proposed methods on logistic regression with simulation study. The class 1 proportion is 0.15 and sample size is 5000.

different correction methods. All three proposed correction methods show large improvement compared to the no correction situation. The predict probability correction method almost always achieves the best performance for classification error rate, specificity, F1 score and G score, but the best sensitivity is always achieved by the data reconstruction method.

5.2 Future work

5.2.1 Different registration error for different covariates

In the thesis we simplified the registration error by assuming the registration errors for different in-vivo measurements were the same. In the real mapping process, different in-vivo measurements may have different registration errors. For example, the mean registration error for histology to T2W was reported to be 1.57 mm, but since there was one more registration step for ADC and DCE, the mean registration error for histology to ADC or DCE was 1.86 mm.

The current weight calculation only takes one registration error, so the prediction is not perfect. One possible solution is to reconstruct different covariates with different registration errors in the data reconstruction method. So for each covariate, the weight is calculated based on its own registration error, and then this covariate is reconstructed based on its own weight.

5.2.2 Weighted loss functions

The SVM classifier can be expressed as an optimization problem that minimizes a loss function subject to some constraints. In the weighted model method, the weighted SVM incorporated the weight in the constraints. It is also possible to construct weighted loss function that incorporates the weight in the loss function, so that reliable instance contributes more in the loss function. This should also help eliminate the impact of the misclassification in response.

5.2.3 Multi-class classification

The prostate cancer is usually labelled as different degrees (i.e. Gleason Scores) in order to distinguish the level of the cancer. In our proposed methods, we treated the cancer status as a binary variable: cancer and non-cancer. If more accurate detection is needed, it would be necessary to do multi-class classification.

For the predict probability correction method, the two misclassification probabilities (2.1)

are exclusive. In order to persist the relationship of the observed class probability and the true class probability, we suggest to use a one-versus-all scheme: when constructing the model for the kth class, treat the rest classes as one label and then perform the predict probability correction method.

The weighted model method and the weighted data reconstruction method are more flexible, so both one-versus-one and one-versus-all schemes can be applied.

Bibliography

Abellán, J. and Masegosa, A. R. (2009). An experimental study about simple decision trees for bagging ensemble on datasets with classification noise. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 446–456. Springer.

Abellán, J. and Masegosa, A. R. (2010). Bagging decision trees on data sets with classification noise. In *International Symposium on Foundations of Information and Knowledge Systems*, pages 248–265. Springer.

Alberts, A. R., Schoots, I. G., and Roobol, M. J. (2015). Prostate-specific antigen-based prostate cancer screening: Past and future. *International Journal of Urology*, 22(6):524–532.

Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D., and Li, S. (2018). *FNN: Fast Nearest Neighbor Search Algorithms and Applications*. R package version 1.1.2.2.

Bonekamp, D., Jacobs, M. A., El-Khouli, R., Stoianovici, D., and Macura, K. J. (2011). Advancements in mr imaging of the prostate: from diagnosis to interventions. *Radiographics*, 31(3):677–703.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Carroll, R. J., Delaigle, A., and Hall, P. (2009). Nonparametric prediction in measurement error models. *Journal of the American Statistical Association*, 104(487):993–1003.

Carroll, R. J., Ruppert, D., Stefanski, L. A., and Crainiceanu, C. M. (2006). *Measurement error in nonlinear models: a modern perspective*. CRC press.

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.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161.

Eckert, R. S., Carroll, R. J., and Wang, N. (1997). Transformations to additivity in measurement error models. *Biometrics*, 53(1):262–272.

Eskin, E. (2000). Detecting errors within a corpus using anomaly detection. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 148–153. Association for Computational Linguistics.

Filson, C. P., Marks, L. S., and Litwin, M. S. (2015). Expectant management for men with early stage prostate cancer. *CA: a cancer journal for clinicians*, 65(4):264–282.

Folleco, A., Khoshgoftaar, T. M., Van Hulse, J., and Bullard, L. (2008). Identifying learners robust to low quality data. In *2008 IEEE International Conference on Information Reuse and Integration*, pages 190–195. IEEE.

Freedman, L. S., Fainberg, V., Kipnis, V., Midthune, D., and Carroll, R. J. (2004). A new method for dealing with measurement error in explanatory variables of regression models. *Biometrics*, 60(1):172–181.

Frénay, B. and Verleysen, M. (2013). Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The Elements of Statistical Learning*, volume 1. Springer series in statistics Springer, Berlin.

Gibson, E., Crukley, C., Gaed, M., Gómez, J. A., Moussa, M., Chin, J. L., Bauman, G. S., Fenster, A., and Ward, A. D. (2012). Registration of prostate histology images to ex vivo mr images via strand-shaped fiducials. *Journal of Magnetic Resonance Imaging*, 36(6):1402–1412.

Gibson, E., Fenster, A., and Ward, A. D. (2013). The impact of registration accuracy on imaging validation study design: A novel statistical power calculation. *Medical image analysis*, 17(7):805–815.

Girosi, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural computation*, 10(6):1455–1480.

Hausman, J. A., Abrevaya, J., and Scott-Morton, F. M. (1998). Misclassification of the dependent variable in a discrete-response setting. *Journal of econometrics*, 87(2):239–269.

Hechenbichler, K. and Schliep, K. (2004). Weighted k-nearest-neighbor techniques and ordinal classification. *Discussion Paper 399, SFB 386, Ludwigs-Maximilians University Munich*, page 16.

Iturria, S. J., Carroll, R. J., and Firth, D. (1999). Polynomial regression and estimating functions in the presence of multiplicative measurement error. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):547–561.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20.

Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*.

Khudyakov, P., Gorfine, M., Zucker, D., and Spiegelman, D. (2015). The impact of covariate measurement error on risk prediction. *Statistics in medicine*, 34(15):2353–2367.

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.

Kuhn, H. (1951). Aw tucker, nonlinear programming. In *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Lin, H., Lin, C., and Weng, R. C. (2007). A note on platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276.

Lindsay, B. (1982). Conditional score functions: Some optimality results. *Biometrika*, 69(3):503–512.

Makarov, D. V., Desai, R. A., James, B. Y., Sharma, R., Abraham, N., Albertsen, P. C., Penson, D. F., and Gross, C. P. (2012). The population level prevalence and correlates of appropriate and inappropriate imaging to stage incident prostate cancer in the medicare population. *The Journal of urology*, 187(1):97–102.

Malley, J. D., Kruppa, J., Dasgupta, A., Malley, K. G., and Ziegler, A. (2012). Probability machines. *Methods of Information in Medicine*, 51(01):74–81.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2019). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.7-0.1.

Michels, K. B. (2001). A renaissance for measurement error. *International Journal of Epidemiology*, 30(3):421–422.

Miranda, A. L., Garcia, L. P. F., Carvalho, A. C., and Lorena, A. C. (2009). Use of classification algorithms in noise detection and elimination. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 417–424. Springer.

Mwalili, S. M., Lesaffre, E., and Declerck, D. (2008). The zero-inflated negative binomial regression model with correction for misclassification: an example in caries research. *Statistical methods in medical research*, 17(2):123–139.

Neuhaus, J. M. (1999). Bias and efficiency loss due to misclassified responses in binary regression. *Biometrika*, 86(4):843–855.

Neuhaus, J. M. (2002). Analysis of clustered and longitudinal binary data subject to response misclassification. *Biometrics*, 58(3):675–683.

Patel, A. R. and Jones, J. S. (2009). Optimal biopsy strategies for the diagnosis and staging of prostate cancer. *Current opinion in urology*, 19(3):232–237.

Pepe, M. S. (1992). Inference using surrogate outcome data and a validation sample. *Biometrika*, 79(2):355–365.

Pepe, M. S., Reilly, M., and Fleming, T. R. (1994). Auxiliary outcome data and the mean score method. *Journal of Statistical Planning and Inference*, 42(1-2):137–160.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Pokorny, M. R., De Rooij, M., Duncan, E., Schröder, F. H., Parkinson, R., Barentsz, J. O., and Thompson, L. C. (2014). Prospective study of diagnostic accuracy comparing prostate cancer detection by transrectal ultrasound–guided biopsy versus magnetic resonance (mr) imaging

with subsequent mr-guided biopsy in men without previous prostate biopsies. *European urology*, 66(1):22–29.

Prentice, R. (1982). Covariate measurement errors and parameter estimation in a failure time regression model. *Biometrika*, 69(2):331–342.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sánchez, J. S., Barandela, R., Marqués, A. I., Alejo, R., and Badenas, J. (2003). Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, 24(7):1015–1022.

Sepanski, J. H. and Lee, L. (1995). Semiparametric estimation of nonlinear errors-in-variables models with validation study. *Journaltitle of Nonparametric Statistics*, 4(4):365–394.

Sexton, J. and Laake, P. (2007). Boosted regression trees with errors in variables. *Biometrics*, 63(2):586–592.

Stefanski, L. A. and Carroll, R. J. (1985). Covariate measurement error in logistic regression. *The Annals of Statistics*, pages 1335–1351.

Stewart, B., Wild, C. P., and others, International Agency for Research on Cancer, W. H. O. (2014). World cancer report 2014.

Sun, J., Zhao, F., Wang, C., and Chen, S. (2007). Identifying and correcting mislabeled training instances. In *Future generation communication and networking (FGCN 2007)*, volume 1, pages 244–250. IEEE.

Teng, C. M. (2000). Evaluating noise correction. In *Pacific Rim International Conference on Artificial Intelligence*, pages 188–198. Springer.

Van Hulse, J. and Khoshgoftaar, T. (2009). Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, 68(12):1513–1542.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition.

Verbaeten, S. and Van Assche, A. (2003). Ensemble methods for noise elimination in classification problems. In *International Workshop on Multiple Classifier Systems*, pages 317–325. Springer.

Ward, A. D., Crukley, C., McKenzie, C., Montreuil, J., Gibson, E., Gomez, J. A., Moussa, M., Bauman, G., and Fenster, A. (2010). Registration of in vivo prostate magnetic resonance images to digital histopathology images. In *International Workshop on Prostate Cancer Imaging*, pages 66–76. Springer.

Ward, A. D., Crukley, C., McKenzie, C. A., Montreuil, J., Gibson, E., Romagnoli, C., Gomez, J. A., Moussa, M., Chin, J., Bauman, G., et al. (2012). Prostate: registration of digital histopathologic images to in vivo mr images acquired by using endorectal receive coil. *Radiology*, 263(3):856–864.

Xu, L., Crammer, K., and Schuurmans, D. (2006). Robust support vector machine training via convex outlier ablation. In *AAAI*, volume 6, pages 536–542.

Yan, Y. and Yi, G. Y. (2016). A class of functional methods for error-contaminated survival data under additive hazards models with replicate measurements. *Journal of the American Statistical Association*, 111(514):684–695.

Yang, X., Song, Q., and Wang, Y. (2007). A weighted support vector machine for data classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(05):961–976.

Yi, G. Y. (2017). *Statistical Analysis with Measurement Error Or Misclassification*. Springer.

Yi, G. Y. and Reid, N. (2010). A note on mis-specified estimating functions. *Statistica Sinica*, pages 1749–1769.

Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210.

Curriculum Vitae

Name:	Kexin Luo
Post-Secondary Education and Degrees:	2014 - 2019 Direct entry Ph.D. in Biostatistics University of Western Ontario London, ON
Honours and Awards:	OICR Biostatistics Training Initiative (BTI) Studentship Award Scholarship 2016-2018
Related Work Experience:	• Teaching Assistant • Research Assistant • Data Science Consultant The University of Western Ontario 2014 - 2019

Publications:

- Weighted methods for the adjustment of misclassification in response, with applications to prostate cancer imaging study. In preparation.
- Correction of predict classification probability for misclassification in response, with applications to prostate cancer imaging study. In preparation.
- A weighted data reconstruction method for measurement error in covariates, with applications to prostate cancer imaging study. In preparation.