공학박사학위논문

Estimating Minority Class Proportion for Class Imbalance and Class Overlap
: Algorithms and Predicting Maladjusted Soldiers

클래스 불균형과 오버랩에서의 소수 클래스 비율 추정
: 알고리즘 및 부적응병사 예측

2015 년 8 월

서울대학교 대학원

산업공학과 데이터마이닝 전공

선 미 선

Abstract

Estimating Minority Class Proportion for Class Imbalance and Class Overlap : Algorithms and Predicting Maladjusted Soldiers

Meessun Sun
Department of Industrial Engineering
The Graduate School
Seoul National University

In the class imbalance and class overlap problem, the direct application of the predictions of the classification model may cause a substantial error. Most classification algorithms aptly focus on the classification of the majority instances while ignoring or misclassifying the minority instances in class imbalance problems. Moreover, emerging challenges, such as an overlap between classes, complicate the solution of the class imbalance to an even greater extent. In situations in which there is an imbalance and overlap between classes where an imperfect classification task can cause great bias in predicting classes, an analytical framework capable of estimating the minority proportion can serve as a more suitable tool, because it provides more reliable information to identify the real-world probability of an instance being the minority. A method for estimating minority proportions can secure its feasibility by providing a good estimate of these proportions for the test, provided a training set exists. The estimation

of the minority proportion can be divided into two categories depending on whether the approach is detailed or aggregated: (1) obtaining the calibrated probability and (2) predicting the prevalence. First, this study proposes a robust calibration technique, Receiver Operating Characteristics (ROC) Binning, to obtain the calibrated probability accurately even if the prevalence of minorities differs in the test and training sets. The new technique uses the True Positive Rate (TPR) and False Positive Rate (FPR), which are insensitive to class skews, and directly reflect the prevalence of minorities in the test set. This method distinguishes between the nature of data distribution within a class value and the effect of the class prevalence by using an ROC curve; thus, it delivers a robust performance with class skews. The effectiveness of the ROC Binning technique was verified by evaluating it together with well-known calibration methods in terms of the Brier Score and Calibration Loss. Among the evaluated calibration methods, the proposed ROC Binning technique was the outstanding front-runner across entire binary-class datasets. An application of the proposed method, ROC Binning and its modification, namely TPR Binning, are used to assess the real-world Military Personality Inventory (MPI) data of the ROK in an attempt to identify those conscripts who are maladjusted. This process is necessary to determine who among them would qualify for exemption from active military service or need special attention. The MPI presents a kind of class imbalance and overlap problem, wherein the majority fulfills active service and the minority is maladjusted, and a conventional classification model is likely to perform poorly. As an alternative, this study applies both ROC and TPR Binning to estimate the calibrated probability, or the maladjusted propor-

tion of persons sharing similar MPI test results. The results we obtained with the real-world MPI dataset confirmed that the suggested method performs well. Second, this study led to the proposal of a simple and effective prevalence estimation method, Similarity-Based Adjusted Count (SAC), to predict prevalence accurately even if the test and training sets have different class distributions. The SAC method is based on an AC method and uses TPR and FPR values of the training instances that are similar to those of the test instances. The proposed SAC adaptively uses part of the whole training set for the estimation of TPR and FPR, whereas conventional Adjusted Count (AC) uses the whole training set for this purpose. The effectiveness of the SAC method was verified by evaluating it and the other AC-based quantification methods using Support Vector Machine or proportionally weighted k-NN as base classification learner in terms of Absolute Error. Among the evaluated prevalence estimation methods, the proposed SAC stood out as the front-runner across all binary-class datasets. Besides, the prevalence estimate was applied in the form of a class prior for obtaining calibrated probability and classifying instances for Bayes classifiers. In Bayesian classification algorithms, prior probability for a class value directly affects the classification decision. The class prior is intrinsically identical to class prevalence in this study. The experimental results indicated that the use of a prevalence estimate enhances to a greater extent the accuracy of the calibrated probabilities and class predictions that were generated compared to other approaches that rely on the class prevalence of the training set, provided that a training set exists of which the positive proportion differs from that of the test set. Additionally, we proposed a correlation-based Gaussian

Bayesian network (CGBN), which is a hybrid (filter- wrapper) Bayesian classification algorithm that considers both classification accuracy and the intrinsic dependence between attributes.

Keywords: Data Mining, Class Imbalance, Class Overlap, Minority Proportion, Calibrated Probability, Prevalence

Student Number: 2007-30167

Contents

Abstract **i**

Contents **viii**

List of Tables **x**

List of Figures **xiii**

Chapter 1 Introduction **1**

1.1 Class imbalance and class overlap problem 4

1.2 Estimating minority class proportion 7

1.3 Overview of this dissertation 11

1.4 Structure of this dissertation 14

Chapter 2 ROC Binning method **17**

2.1 Background 17

2.2 Related work 21

2.2.1 Calibration method based on the existing classifier 21

2.2.2 Evaluation measures for calibration methods 24

2.2.3 ROC curve 26

2.3 ROC Binning 30

2.4 Performance on benchmark datasets 37
2.4.1 Experiment settings 37
2.4.2 Experiment results 45
2.5 Summary 57

Chapter 3 Application of ROC-based Binning for predicting maladjusted soldiers 61

3.1 Background 61
3.2 Military Personality Inventory Data 63
3.3 Performance of classification algorithms using the MPI dataset . 67
3.4 Use of ROC curve to obtain calibrated probability 69
3.4.1 Generation of ROC curve 70
3.4.2 Obtaining calibrated probability 72
3.5 Summary 84

Chapter 4 Similarity-based Adjusted Count (SAC) method 87

4.1 Background 87
4.2 Related work 90
4.3 Similarity-based adjusted count 95
4.4 Performance on benchmark datasets 101
4.4.1 Experiment settings 101
4.4.2 Experiment results 103
4.5 Summary 113

Chapter 5 Application of prevalence estimate into Bayesian classifier **117**

5.1 Background 117

5.2 Related work 119

5.2.1 BN learning from the undirected graphical model 119

5.2.2 Filter and wrapper approaches 126

5.2.3 Covariance constraint for Gaussian distribution 128

5.3 Finding the best CGBN 130

5.3.1 Step 1: Calculate $\overline{R_c^2}$ to measure the dependence 131

5.3.2 Step 2: Generate CGBNs using SLHC based on $\overline{R_c^2}$ 134

5.3.3 Step 3: Search for bCGBN using z-SEI and FS 136

5.3.4 Applying BSE to bCGBN 139

5.4 Performance on benchmark datasets 141

5.4.1 Experimental settings 141

5.4.2 Experimental results 142

5.5 Summary 144

Chapter 6 Conclusion **147**

6.1 Contributions 147

6.2 Future work 151

Bibliography **155**

Appendix A: First and second PCs of the benchmark datasets **175**

Appendix B: BS distribution per binary dataset by changing prevalence **183**

Appendix C: AE distribution per binary dataset by changing prevalence **191**

Appendix D: Application of SAC into MPI dataset **199**

국문초록 **203**

List of Tables

Table 1.1 Organization of this dissertation 16

Table 2.1 Confusion matrix 27

Table 2.2 Summary of datasets 39

Table 2.3 Summary of recomposed binary-class datasets 41

Table 2.4 BS for each binary-class dataset (e.g., NB) 46

Table 2.5 Summary of BS for all the test conditions 49

Table 2.6 Summary of CL for all the test conditions 53

Table 2.7 BS per test condition (e.g., NB) 55

Table 3.1 Measures and numbers of relevant questions in the MPI . 65

Table 3.2 Performance of classification algorithms applied to the MPI dataset 69

Table 3.3 Comparison of AUC values of classification algorithms on the MPI dataset. 71

Table 3.4 BS for all the test conditons on the MPI dataset 73

Table 3.5 BS per test conditon on the MPI dataset (e.g., LDA) . . . 75

Table 3.6 BS of TPR Binning for all the test conditions on MPI dataset 77

Table 3.7 BS of TPR Binning per test condition on MPI dataset (e.g., LDA) 79
Table 3.8 MSEs and MAEs of the maladjusted proportion of TPR Binning on the MPI dataset 80
Table 3.9 Cumulative % of the maladjusted subjects and subjects in the training and test sets on the MPI dataset 81
Table 3.10 Comparison of the maladjusted proportion with the maladjusted probability p($\mathbf{x}$=maladjusted) on the MPI dataset 84

Table 4.1 AC-based imbalance tolerant methods via threshold selection 94
Table 4.2 AE for each binary-class dataset 104
Table 4.3 Summary of AE for total test conditions 107
Table 4.4 AE per test condition 110

Table 5.1 Strength of CGBN 130
Table 5.2 r_c^2 and r^2 of the Iris dataset 133
Table 5.3 Summary of BS of the calibration methods including ROC binning which use prevalence estimate 143
Table 5.4 Classification accuracy and Wilcoxon signed-rank test result at α=0.05 144

Table 6.1 Methods and applications covered in this dissertation . . . 148

List of Figures

Figure 1.1 Concept of classification model 2

Figure 1.2 General approach for building a classification model . . . 2

Figure 1.3 Class imbalanced data with different class overlap 6

Figure 1.4 Bias in the prediction of classifier 6

Figure 1.5 Categorization of minority proportion 8

Figure 1.6 Usefulness of being able to identify the minority proportion 9

Figure 2.1 Toy example of ROC curve generation from a finite set of instances 30

Figure 2.2 Influence of change in prevalence on calibrated probability 31

Figure 2.3 Changes of ΔTPR, ΔFPR, and positive proportion on the ROC curve 33

Figure 2.4 Procedure used by ROC Binning 34

Figure 2.5 Computation algorithm utilized by ROC Binning 36

Figure 2.6 Box-plots of BS for all the test conditions 50

Figure 2.7 Nemenyi post-hoc test for all the test conditions at α=0.05 52

Figure 2.8 Nemenyi post-hoc tests per test scenario at α=0.05 (e.g., NB) 56

Figure 3.1 Distributions of the subjects (histograms) and maladjusted proportions (lines) of TPR Binning 66
Figure 3.2 First and second PCs of the MPI dataset 67
Figure 3.3 ROC curves of classification algorithms applied to the MPI dataset. 70
Figure 3.4 BS on the MPI dataset 74
Figure 3.5 Computation algorithm utilized by TPR Binning 78
Figure 3.6 Distributions of the subjects (histograms) and maladjusted proportions (lines) of TPR Binning on the MPI . 82

Figure 4.1 Differences in classifier performance by data location . . 96
Figure 4.2 Procedure used by SAC 98
Figure 4.3 Computation algorithm utilized by SAC 100
Figure 4.4 Box-plot of AE 107
Figure 4.5 Nemenyi post-hoc tests for total test conditions at α=0.05 108
Figure 4.6 Nemenyi post-hoc tests per test condition α=0.05 112

Figure 5.1 Example of different BNs 120
Figure 5.2 DAGs with d nodes at iteration i in Eq. (5.3) Red numbers denote redundant DAG identifiers at each d 122
Figure 5.3 Computation algorithm for finding bCGBN 131
Figure 5.4 Distribution of attributes for each class value in the Iris dataset 133
Figure 5.5 Generation of CGBNs using the Iris dataset 135

Figure 5.6 Searching for the SLHC stage corresponding to bCGBN: (a) z-SEI search, (b) Fibonacci search 137

Figure 5.7 Process by which BSE is applied to bCGBN 140

Chapter 1

Introduction

Classification is a pervasive problem that encompasses many diverse applications and it is one of the most important tasks in data mining. A classification task involves the assignment of an input pattern to one of the predefined categorical class labels (Duda et al., 2000; P.-N. Tan et al., 2006). In binary classification, we are given (a realization of) a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, ..., N\}$, where $\mathbf{x}_i$ is an arbitrary attribute set (often a vector in R^d) and $y_i \in \{0, 1\}$ corresponds to a binary-valued class label. The goal of binary classification is to find a decision function whose level sets may be used to separate $\mathbf{x}_i$ from differing class labels y_i. A classification model is learned from a set of training instances with class labels. As shown in Figure 1.1, a classification model can be treated as a black box that automatically assigns a class label when presented with the attribute set of an unknown class label.

A classification algorithm is a systematic approach that is used to learn the classification model from an input dataset. Examples include Naïve Bayes (NB) (Duda & Hart, 1973), Bayesian Quadratic Discriminant Analysis (QDA)/LDA (Switzer, 1980), Logistic Regression (LR)(Agresti, 1996), and Support Vector Machine (SVM) (Cortes & Vapnik, 1995). The performance of a classification

Figure 1.1: Concept of classification model

model is typically measured by its classification accuracy. Thus, a key objective of the classification learning algorithms is to build models with good generalization capability; i.e., models that accurately predict the class labels for records with unknown class labels. Figure 1.2 shows a universal approach for solving classification problems. First, a training set consisting of records whose class labels are known must be provided. The training set is used to build the classification model, which is subsequently applied to the test set, which consists of records with unknown class labels.

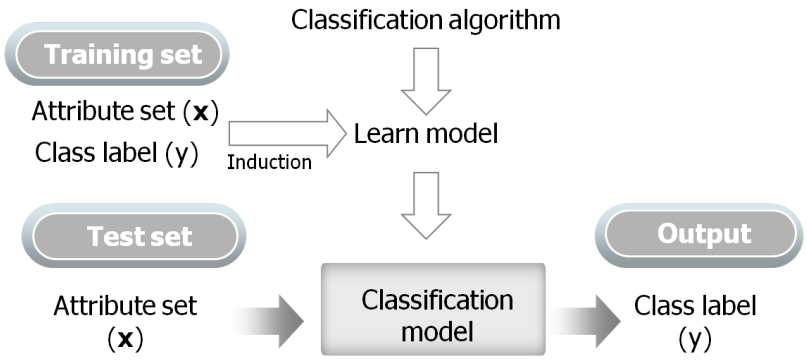


Figure 1.2: General approach for building a classification model

In recent years, the class imbalance and class overlap problem has received considerable attention in classification because it has been recognized in many practical domains, but it causes suboptimal classification performance and most conventional algorithms behave badly in those situations. Many management, business, and diagnostic fields would benefit from the ability to identify the typical patterns of a target class in situations in which class imbalance and class overlap occur. For example, the Military should identify minority class instances such as maladjusted conscripts to determine who among conscription subjects or the ranks are qualified for exemption from active military service or need special attention during active military service. Similar situations in which the minority class instances need to be identified are observed in other areas, such as risk prediction in business, fraud detection in banking operations, fault diagnosis, medical diagnosis, and image recognition. Many difficult real-world issues are characterized by class imbalance and class overlap problems (Burez & Poel, 2009; Galar et al., 2012; Shin & Cho, 2006; Weiss, 2004; Xiao et al., 2012).

In class imbalance and class overlap problems, the direct application of the prediction of a classification model could result in substantial error, because most classification algorithms are apt to focus on the classification of the majority instances while ignoring or misclassifying the minority instances (Duman et al., 2012). Alternatively, the classification model may predict every instance as a major class in the class imbalance and overlap problem. Thus, techniques are required to solve the problems associated with class imbalance.

1.1 Class imbalance and class overlap problem

Class imbalance is a situation where one class is represented by a large number(N.) of instances, whereas the other is represented by only a few. In class imbalance problems, it is quite challenging to use conventional classification algorithms to build well performing classifiers(Japkowicz & Stephen, 2002; Lee, 2007; Weiss, 2004). Thus, class imbalance is likely to decrease the performance of classification algorithms (Denil & Trappenberg, 2010).

Several reasons have been mentioned for class imbalance, which make classification difficult (Japkowicz & Stephen, 2002; Lee, 2007). First, it is difficult to detect regularities for the minority class when the instances of a minority class are very rare in an absolute sense. Second, the minority class can be ignored by classification algorithms in a relative sense if the instances of the minority class may be relatively rare compared with those of the majority class (Buntine, 1992). Third, classification algorithms become highly vulnerable to noise in class imbalance problems. Fourth, the accuracy may be meaningless or even misleading for a class imbalance program, which should put a premium on the minority class, since conventional classification algorithms count both classes of instances equally.

Previous studies proposed a number of solutions to class imbalance problems at data or algorithmic levels (Barandela et al., 2003; Chawla et al., 2004; Das et al., 2014; Kotsiantis et al., 2006). At the data level, these solutions include many different forms of re-sampling such as over sampling (i.e., adding new minority samples) and under sampling (i.e., removing exiting majority samples)

(Batista et al., 2004; Chawla et al., 2002; Liu et al., 2009). These approaches might cause problems in terms of either over fitting or losing potentially useful information, respectively. At the algorithmic level, solutions include adjusting the costs of the various classes, adjusting the decision threshold, and novelty-based learning (from one class) rather than discrimination-based learning (from two classes), such as to counter class imbalance (Lee, 2007; Y. Sun et al., 2007). These approaches are likely to obtain higher prediction accuracy for minority classes at the expense of lowering the prediction accuracy for the majority class. Thus, either conventional classification algorithms or approaches focusing on class imbalance are likely to distort the actual occurrences of the classes practically.

Emerging challenges associated with class overlapping makes class imbalance even harder to solve. The class overlap problem presents ambiguous regions in feature space in which there are an approximately similar number of training instances from different classes (Denil & Trappenberg, 2010). In cases such as this, instances sharing similar attribute values might belong to different classes. Recent findings have shown that class overlap can play an even larger role in classifier performance than class imbalance (Das et al., 2014; Prati et al., 2004). Experimental results with SVM and k-NN showed that, in the presence of an imbalance with low class overlap, the classifiers provide high performance on both classes (Das et al., 2014; García et al., 2006, 2008). Figure 1.3 visually illustrates the difference between datasets with a separate class boundary and those with class overlap. The combination of class imbalance and class overlap complicates the classification problem (Alejo et al., 2013; García et al., 2006).

(a) data with a separate class boundary (b) data with class overlap

Figure 1.3: Class imbalanced data with different class overlap

Both conventional classifiers and imbalance tolerant classifiers are likely to provide the biased class predictions on imbalanced and overlapped data as shown in Figure 1.4. Conventional classifiers make more majority predictions than they really are, whereas imbalance tolerant classifiers make more minority predictions than they really are. Especially the instances located in a class overlap region largely depend on which classifier is employed.

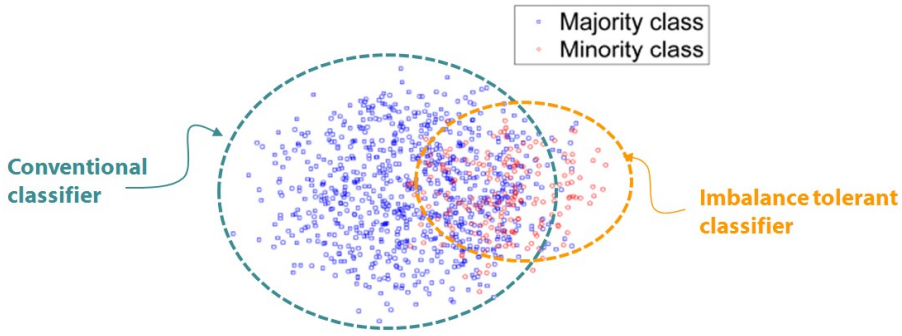


Figure 1.4: Bias in the prediction of classifier

1.2 Estimating minority class proportion

In situations in which a classifier is required to identify the corresponding class labels for given patterns, an imperfect classification task can cause substantial bias in class prediction. Instead, an analytical framework capable of estimating the minority proportion could serve as a more suitable tool, because it would provide more reliable information to identify the real-world probability of an instance being a minority. Methods with the capability of providing a good estimation of the minority proportion of a test set could secure the feasibility of the framework, provided the training set contains a minority proportion that is noticeably different from that in the test set (Barranquero et al., 2013).

Estimation of the minority proportion can be divided into two categories depending on whether a partitioning or aggregation approach is followed: (1) obtaining the calibrated probability and (2) predicting prevalence, respectively, as shown Figure 1.5. The use of a calibrated probability enables us to identify the real-world probability of being a minority for individual instances or subsets, whereas the use of prevalence prediction enables us to predict the difference in the total minority proportion between different datasets.

The decision on management can be made in terms of profit maximization using the calibrated probability and the prevalence. We may use an example from the Military Personality Inventory (MPI), which is used to identify maladjusted conscripts. In the Military, the MPI test is currently used to select approximately 7% of the conscription subjects per year for further psychiatric diagnosis to determine their eligibility for exemption from military duty. If the

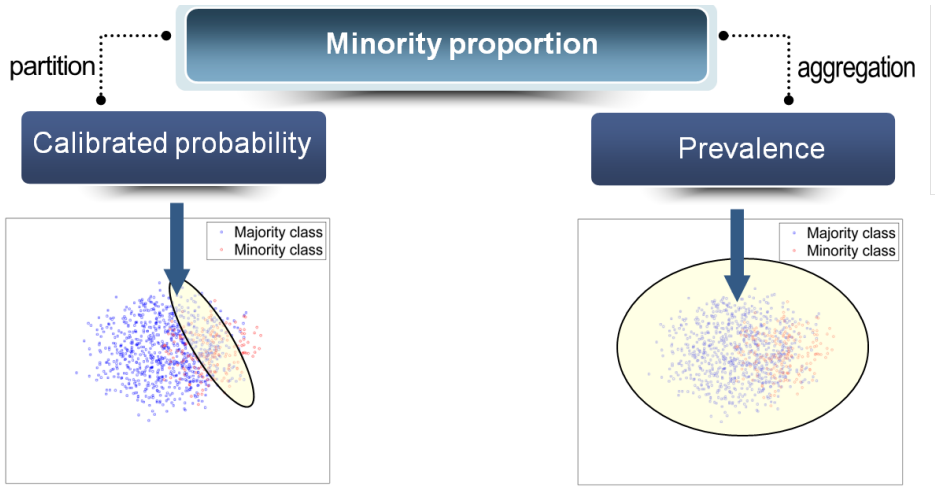


Figure 1.5: Categorization of minority proportion

calibrated probability indicates that the top 7% of the subject scores comprise 25% of those considered maladjusted, a maximum of 25% of the maladjusted can be exempted from active military service in advance. If subject subsets of greater maladjusted proportions were to enter the military service, they would have to be provided with additional attention during service with sufficient priority (M. Sun et al., 2015). Prevalence estimation can be used to predict differences in the maladjusted proportion among different corps/divisions. If the national Ministry of Defense were to identify a division with a higher prevalence, it would have to instruct the commander of the division to proactively prevent the maladjusted problem (e.g., counseling, mentoring, and education) and would have to reassign some of the maladjusted to another division. This kind of decision making is observed in other situations, such as cost/loss analy-

sis, customer value analysis, alternative analysis, market research, and incidence prediction as shown Figure 1.6. If we have a number of alternatives in decision making, we would be able to determine the relative and absolute priorities in terms of cost-effectiveness based on the information obtained by determining the calibrated probability and prevalence.

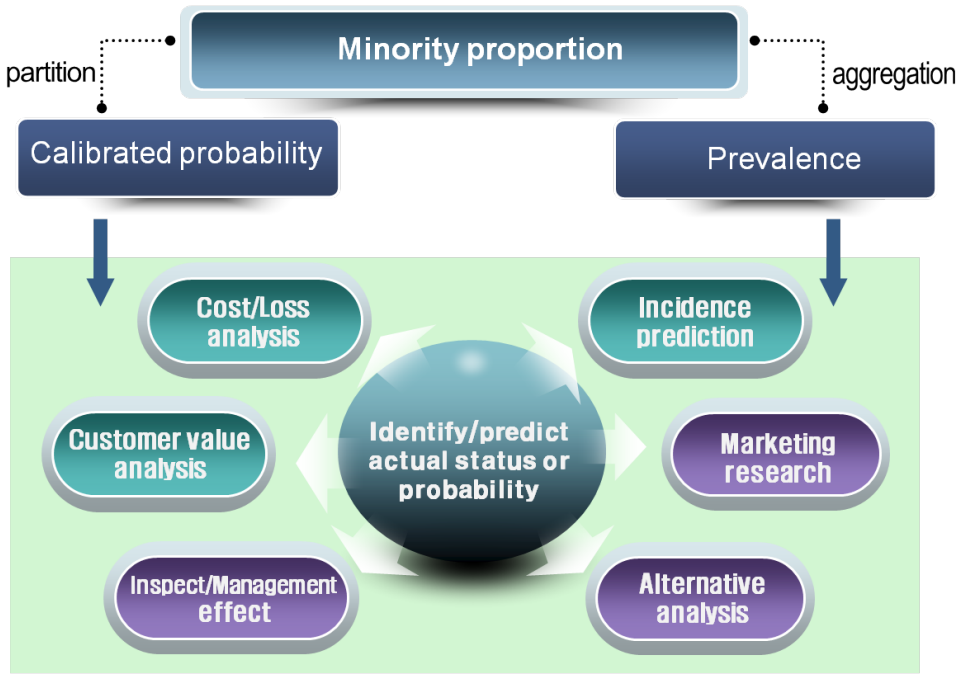


Figure 1.6: Usefulness of being able to identify the minority proportion

Obtaining the calibrated probability, or the actual occurrence, is a significant issue in the class imbalance and class overlap problem and is crucial in many real applications because it supports decision making with the precise

assessment of cost and effect (Zadrozny & Elkan, 2001). Calibrated probabilities are defined as the degree of approximation of the estimated probabilities to the actual probabilities (Bella et al., 2013). If we predict that we are 80% sure that a specific event will occur, we expect the occurrence proportion of the event to be right 80% of the time. More precisely, we can say that a probability is perfectly calibrated if, for a bin or a subset of instances with the estimated probability p of being a positive class, the expected proportion of positives in the class should be equal to p.

Predicting the prevalence of a target dataset is required to track trends over time and identify the difference between different targets. The aim of prevalence prediction is to estimate the total minority proportion of a test set, provided that we have a training set in which this total minority proportion may be noticeably different (Barranquero et al., 2013). It is sufficient, but not necessary, to have a perfect classifier to estimate the prevalence comparatively well (Forman, 2008). Intuitively, the prevalence estimation task is easier, in that it does not require the delivery of accurate predictions on the individual level, because the statistical nature of the uncertainty is shifted from the individual case to the aggregate count of cases.

And besides, estimation of the minority proportion can be used to consider the suitability of an existing classifier. When the calibrated probability or the estimated prevalence is too different from that of the training set, the existing classifier may provide poor classification performance. In that case, we need to rebuild a new classifier.

1.3 Overview of this dissertation

In summary, the main contributions of this dissertation are four-fold as follows. First, this study proposes the robust calibration technique, Receiver Operating Characteristics (ROC) Binning, which can be used to obtain the calibrated probability accurately even when the test and training sets have a different prevalence of positives. Obtaining the calibrated probability is an effective way to deal with class imbalance and class overlap problems. The motivational intuition behind this work is that the inherent difference in the prevalence of positive class between the training and test sets of interest could largely affect the performance of the calibration method, especially when classes of data are imbalanced and overlapped. Previous studies dealing with calibrated probability barely took account of the difference in class prevalence between the training and test sets. The proposed method can provide the accurate calibrated probability by distinguishing the distribution nature within a class value and the influence of the prevalence of the positive class. This study uses the True Positive Rate (TPR) and False Positive Rate (FPR), which are insensitive to class skew, as performance measures for the classifier, to generate bins for calibration. Moreover, the positive fraction of each bin is used directly as the calibrated probability to avoid probability distortion. The proposed approach is expected to improve the accuracy of the calibrated probability it generates in comparison to other approaches that do not take into account the change in class skew, provided that we have a training set in which the positive proportion may be noticeably different from that of the test set.

Second, ROC Binning and its modification are applied into the real-world MPI dataset of the ROK. ROK Armed Forces include maladjusted conscripts, such as the mentally ill, the suicidal, the imprisoned, and those determined by the military commander to be maladjusted. To counteract these problems, MPI is used to identify the maladjusted conscripts to determine who among them would qualify for exemption from active military service or need special attention. Therefore, MPI presents a kind of class imbalance and class overlap problem, in that the majority fulfill active service and the minority are maladjusted, and its classification result is likely to show poor performance. As an alternative, this study employs ROC Binning and its modification to estimate the calibrated probability, or the maladjusted proportion of persons sharing similar MPI test results. Experiments are conducted to validate that the suggested method performs well for the real-world MPI dataset.

Third, this study suggests the Similarity-based Adjusted Count (SAC) to estimate prevalence by reflecting the difference in data distribution between the training and test sets. The objective is to accurately estimate the prevalence of positive class in the test set, provided that we have a training set in which this distribution may be different from that in the test set. SAC is an adaptive prevalence estimation method, derived from the Adjusted Count (AC) method, using the TPR and FPR of training instances similar to test instances. This method fundamentally assumes that a change in the data distribution between the training and test sets could affect the TPR and FPR; however, similar instances in the feature space would share fairly similar values of the TPR and FPR. This study uses the k-NN model as base classification learner, because

this model is able to remember details of the data topology and search similar instances that are intrinsically based on distance. The proposed approach is expected to improve the accuracy of the prevalence estimate compared with other approaches that do not take into account the difference in data distribution, provided that we have a training set in which the positive proportion may be noticeably different from that of the test set.

Finally, the result of the prevalence estimate result is applied as class prior for obtaining the calibrated probability and classifying instances using Bayes classifiers. A Bayes classifier can be represented as a Bayesian network among attributes and class. Bayesian classification algorithms allow us to express the posterior probability of each class conditioned on an instance in terms of the evidence, class conditional probability, and prior probability for the class value. Thus, the prior probability for a class value directly affects the classification decision. The class prior is intrinsically identical to class prevalence in this study. The use of the prevalence estimate as class prior knowledge is expected to enhance the accuracy of calibrated probabilities and class predictions in relation to other approaches that rely on a class prevalence of the training set, provided that we have a training set of which the positive proportion may be noticeably different from that of the test set. Additionally, we proposed a correlation-based Gaussian Bayesian network (CGBN), a hybrid Bayesian classification algorithm, which considers both classification accuracy and intrinsic dependence between attributes.

1.4 Structure of this dissertation

This dissertation suggests a methodology and its application that aim to accurately estimate the minority proportions of a test set for binary-classification problems with class imbalance and class overlap. First, a systematic method for obtaining the calibrated probability, ROC Binning, is proposed and applied to real-world datasets. ROC Binning is a robust method designed to obtain an accurate calibrated probability even with changes in the minority prevalence. ROC Binning is capable of distinguishing the distribution nature within a class value and the influence of the prevalence. Additionally, the proposed method is applied to real-world UCI datasets and an MPI dataset. MPI is used by the Military to identify maladjusted conscripts who qualify for exemption from active military service or need special attention during military service. MPI presents a kind of class imbalance and overlap problem, which requires an effective way to identify the real-world probability of being maladjusted.

Second, a prevalence estimation method, Similarity-based Adjusted Count (SAC), is proposed and applied to real-world datasets. SAC is an adaptable method that can be used to change the test distribution. SAC estimates prevalence by using the True Positive Rate (TPR) and False Positive Rate (FPR) of the classifier for training instances similar to test instances, enabling us to take into account the locality or the topology of test instances. The proposed method is applied into real-world UCI datasets. Additionally, we apply the prevalence estimate result as class prior for obtaining the calibrated probability and classifying instances by Bayes classifiers.

This dissertation is structured as in Table 1.1 and organized as follows. This chapter introduces the concept and the need for estimating the minority proportion in situations of class imbalance and class overlap. In Chapter 2, the previously introduced methods for obtaining the calibrated probability are briefly reviewed. Then, the ROC Binning method for binary-class problems is proposed and its performance is tested and compared with other popular calibration methods. In Chapter 3, the ROC Binning method and its modification are applied to the real-world MPI dataset of the ROK. This chapter demonstrates that the suggested method is very useful to estimate the proportion of conscripts who are likely to be maladjusted to military life and would be able to provide assistance with the management of the persons subject to conscription or the ranks. In Chapter 4, the previously introduced methods for prevalence estimation are briefly reviewed. Then, the SAC method for binary-class dataset is proposed and its performance is tested and compared to other prevalence estimation methods. In Chapter 5, prevalence estimate is applied as class-prior for calibration and classification by various Bayes classifiers including the newly proposed Gaussian Bayesian network. Finally, Chapter 6 concludes the dissertation by presenting the limitations of the proposed approaches and related future work.

Table 1.1: Organization of this dissertation

Chapter	Description	
Chapter 1	Introduction	-
Chapter 2	Part Ⅰ: Obtaining calibrated probability	ROC Binning method
Chapter 3		Application of ROC-based Binning for predicting maladjusted soldiers
Chapter 4	Part Ⅱ: Predicting prevalence	Similarity-based Adjusted Count (SAC) method
Chapter 5		Application of prevalence estimate into Bayes classifier
Chapter 6	Conclusion	-

Chapter 2

ROC Binning method

2.1 Background

Obtaining calibrated probability, or actual occurrence proportion, is crucial in many real-world problems because it effectively supports the decision-making process with good assessment of cost and effect. Calibrated probability is defined as the degree of approximation of the predicted probabilities to the actual probabilities (Bella et al., 2013). Thus, if we predict that we are 80% sure, we should expect to be right 80% of the time. More precisely, a classifier is perfectly calibrated if, for a bin or a subset of instances with predicted probability p for the positive class, the expected proportion of the positive class should be equal to p. Producing well-calibrated probabilistic prediction is crucial in many areas (Naeini et al., 2014). For example, it is required in science (e.g., when determining which experiment to perform), medicine (e.g., when deciding which therapy to give a patient), management (e.g., when selecting priority control targets), and business (e.g., when developing an investment plan), for decision making in terms of expected cost or benefit.

Estimating calibrated probability is a very significant issue, especially in

class imbalance and class overlap problems where direct application of classification model predictions can result in substantial errors. In class imbalance and overlap problems, conventional classification algorithms usually focus on prediction of the majority instances while ignoring or misclassifying the minority instances, which results in poor performance. An analytical framework that estimates the actual probability of the minority class can serve as a more suitable tool in this stead, because it provides more granular information. A calibrated probability offers a realistic impression of the true probability that each test pattern is a member of the class of interest (Cohen & Goldszmidt, 2004). The calibrated probabilities are intuitively interpretable and useful for estimating expected cost and effect, and generally offer more aggregate and concrete information than class predictions alone.

In a classification dataset that contains categorically dependant variables, the value of the function from the classifier can represent degrees of confidence for choosing the class label, thereby accompanying the class prediction with a likelihood score. One way to achieve a high level of calibration is to develop probabilistic models that are well-calibrated. However, this approach requires modification of the objective function used in the classifier to learn the probabilistic model and it may increase the computational cost of the associated optimization task (Lambrou et al., 2012; Naeini et al., 2014). For this reason, instead of redesigning the existing classifiers to obtain probability values, several calibration techniques obtain well-calibrated probabilities by relying on the existing classifier as is (Naeini et al., 2014). Further, this approach can be used in various kinds of classification algorithms, and is often preferred because of its

generality and flexibility, and the fact that it frees the designer of the machine learning model from the need to add extra calibration criteria to the objective function used to learn the probabilistic model.

Several calibration techniques have been developed for these purposes, with some of them, such as Histogram Binning (Zadrozny & Elkan, 2001), Platt Scaling (Platt, 1999), and Isotonic Regression (Zadrozny & Elkan, 2002), being comparatively well-known. In these cases, calibration techniques are developed as post-processing techniques with the aim of improving the probability estimation or the error distribution of an existing classification model. This post-processing step can be regarded as a function that maps the outputs of the prediction model to the probability values that are intended to be well-calibrated.

The motivational intuition beyond this work is that the inherent difference in the prevalence of positive class, or total minority proportion, between the training and test sets can affect the performance of the calibration method, especially when the classes of data are imbalanced and overlapped. The calibrated probability is obtained in the context of the training set (Agoritsas et al., 2011), and the probability of being positive is directly affected by the prevalence of the positive class in the training set. If the prevalence of the positive class in the test set is different from that in the training set, the instance among the specific classifier's scores could also have a different positivity probability.

This study proposes Receiver Operating Characteristics (ROC) Binning, a novel method that obtains accurate calibrated probabilities that are robust to changes in the prevalence of the positive class. The proposed method can

robustly provide accurate calibrated probabilities by differentiating the distribution nature within a class value from the influence of the prevalence of the positive class. The method uses True Positive Rate (TPR), False Positive Rate (FPR), and the performance measures of the classifier (which are insensitive to class skew), to generate bins for calibration. Further, to avoid probability distortion, the positive fraction for each bin is used directly as the calibrated probability. To verify the effectiveness of the proposed ROC Binning method, we conducted comparative evaluations with well-known calibration methods in terms of Brier Score and Calibration Loss. Among the calibration methods evaluated, the proposed ROC Binning proved superior across all binary-class datasets.

This chapter is organized as follows. In Section 2.2, we briefly review previous work related to calibration methods that rely on the existing classifier. In Section 2.3, we demonstrate the proposed ROC Binning method, which is robust to changes in test prevalence. In Section 2.4, experimental settings such as experimental methodology, datasets, classification algorithms, and calibration methods are discussed and the experiments conducted outlined.And we analyze the experimental results obtained. Finally, we discuss the main conclusions and possible future research paths in Section 2.5.

2.2 Related work

2.2.1 Calibration method based on the existing classifier

Histogram Binning (Zadrozny & Elkan, 2001) is the simplest method for calibration that uses functional values from a classifier. Recall the setup for a binary classification problem, in which we are given (a realization of) a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, ..., N\}$, where $\mathbf{x}_i$ is an arbitrary attribute set (often vectors in R^d) and $y_i \in \{0, 1\}$ is the corresponding binary-valued class label. It sorts the training instances $(\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots\ , \mathbf{x}_N)$ according to their scores and divides the sorted set $(\mathbf{x}_{(1)},\ \mathbf{x}_{(2)},\ \ldots\ , \mathbf{x}_{(N)})$ into a predefined number of subsets, M, of equal size, called bins. For each bin, b_k, it checks the lower and upper bound scores and counts the total number of instances $n_{(b_k)}$ and number of positives $n_{1(b_k)}$. Here, in the general class imbalance problem, the minority and majority are designated as the positive class and the negative class, respectively. In the test phase, Histogram Binning assigns each test instance to a bin, b_k, where $1 \leq k \leq M$, according to its score. Finally, the calibrated probability of a class being a positive class for instances belonging to each bin is estimated using a fraction of the positives of the assigned bin in the training set. The Histogram Binning method for obtaining calibrated probability, $\hat{p}_{HB}(\mathbf{x}_{b_k})$, is formulated as in Eq. (2.1):

$$\hat{p}_{HB}(\mathbf{x}_{b_k}) = \frac{n_{1(b_k)}}{n_{(b_k)}}, \forall \mathbf{x}_{b_k} \in b_k. \tag{2.1}$$

Instances belonging to the same bin share identical calibrated probability. Histogram Binning is a parametric calibration method because the number of bins

must be predefined in order to make equal sized bins.

In the class imbalance problem, several bins may be fulfilled with only the majorities, or the negative instances, in the class imbalanced dataset. This is caused primarily by the class prevalence, not by the distribution nature of the within class value. In such a case, the calibrated probability obtained by Histogram Binning is useless because the class imbalance problem requires sensitive information for the minorities (Wallace & Dahabreh, 2012).

Platt (Platt, 1999) presented a parametric approach for fitting a sigmoid function that maps a classifier's scores to calibrated probabilities; this method is known as Platt Scaling or Platt logistic regression. The main task is to determine the parameters of the sigmoid function that minimizes the negative log-likelihood of the data using a classifier's score $f(\mathbf{x}_i)$ for each instance $\mathbf{x}_i$. The calibrated probability obtained by the Platt Scaling method, $\hat{p}_{PS}(\mathbf{x}_i)$, is formulated as

$$\hat{p}_{PS}(\mathbf{x}_i) = \frac{1}{1 + exp(\beta_0 + \beta_1 f(\mathbf{x}_i))}. \tag{2.2}$$

Parameters β_0 and β_1 are estimated using a maximum likelihood method that optimizes the parameters on the training set. To avoid overfitting the training set, Platt additionally suggested transforming the y labels to target probabilities for positive and negative instances, $t_{c=1}$ and $t_{c=0}$, respectively (Eqs. (2.3) and (2.4))

$$t_{c=1} = \frac{n_1 + 1}{n_1 + 2} \tag{2.3}$$

$$t_{c=0} = \frac{1}{n_0 + 2} \tag{2.4}$$

where n_1 and n_0 represent the number of positives and negatives, respectively. Platt further suggested using the Levenberg–Marquardt algorithm to optimize the parameters. However, a Newton algorithm that was later proposed by Lin et al. (Lin et al., 2007) became more popular because it is deemed to be more numerically stable.

In the Isotonic Regression method (Zadrozny & Elkan, 2002), the calibrated probabilities are obtained via an isotonic (monotonically increasing) mapping function. The Pool Adjacent Violators (PAV) algorithm is popularly used to fit the training set to that function (Fawcett & Niculescu-Mizil, 2007). The initial step in this algorithm sorts the instances in decreasing order according to their score and initializes $p^*(\mathbf{x}_i) \leftarrow y_i$, where $y_i \in \{0, 1\}$. The underlying idea is that calibrated probability estimates must be a decreasing sequence, i.e., $p^*(\mathbf{x}_{(1)}) \geq p^*(\mathbf{x}_{(2)}) \geq \ldots \geq p^*(\mathbf{x}_{(N)})$. If this is not the case, then for each pair of consecutive probabilities, $p^*(\mathbf{x}_{(i)})$ and $p^*(\mathbf{x}_{(i+1)})$, such that $p^*(\mathbf{x}_{(i)}) < p^*(\mathbf{x}_{(i+1)})$, the PAV algorithm replaces both of them by their probability average:

$$p^*(\mathbf{x}_{(i)}) \leftarrow \frac{p^*(\mathbf{x}_{(i)}) + p^*(\mathbf{x}_{(i+1)})}{2},\ p^*(\mathbf{x}_{(i+1)}) \leftarrow \frac{p^*(\mathbf{x}_{(i)}) + p^*(\mathbf{x}_{(i+1)})}{2}. \quad (2.5)$$

This is a nonparametric method that leads to a stepwise constant mapping function. Isotonic Regression employs the basic model defined in Eq. (2.6):

$$\hat{y}_i = \hat{p}_{IR}(\mathbf{x}_i) + \varepsilon_i \quad (2.6)$$

where $\hat{p}_{IR}(\mathbf{x}_i)$ is a non-decreasing mapping function of the calibrated probability by the Isotonic Regression method and can be found with the mean square error

criterion of Eq. (2.7):

$$\begin{aligned} &\min_{\hat{p}_{IR}} \sum_{i=1}^{N} (\hat{p}_{IR}(\mathbf{x}_i) - y_i)^2 \\ &s.t.\ \hat{p}_{IR}(\mathbf{x}_i) \geq \hat{p}_{IR}(\mathbf{x}_j)\ all(i,j) \in E. \end{aligned} \tag{2.7}$$

Platt Scaling and Isotonic Regression convert the classifier's score into a specific function model. However, this conversion is vulnerable to noise and is highly dependent on the classifier's performance. Platt Scaling is particularly effective when the actual probabilities are sigmoid-shaped, whereas Isotonic Regression, which naturally generates a monotonic function, is usually a more powerful calibration method. However, learning curve analysis has shown that Isotonic Regression is more prone to overfitting, and thus performs worse than Platt Scaling when data are scarce (Niculescu-Mizil & Caruana, 2005).

Many calibration methods have been presented in previous studies(Gebel, 2009). However, of these calibration methods, the ones conventionally used are Histogram Binning, Platt Scaling, and Isotonic Regression. This study empirically compares the proposed ROC Binning method to these conventionally used calibration methods.

2.2.2 Evaluation measures for calibration methods

Calibration quality can be evaluated by measures used for classification because a perfectly calibrated classifier should predict the probability of being positive as a discrete one or zero (Ferri et al., 2009). Brier Score, or Mean Squared Error (MSE) for binary classes, penalizes deviations from the true probability (Brier, 1950). For the binary classification case, the Brier Score is defined as the average squared difference between the estimated probabilities of being positive

and the true class labels, zero (negative class value) or one (positive class value), for instances and is formulated as

$$\text{Brier Score} = \frac{\sum_{i=1}^{n}(\hat{p}(Y=1/\mathbf{x}_i) - y_i)^2}{N} = \frac{\sum_{i=1}^{n}(\hat{p}(\mathbf{x}_i) - p(\mathbf{x}_i))^2}{N} \tag{2.8}$$

where $\hat{p}(Y=1/\mathbf{x}_i)$ or $\hat{p}(\mathbf{x}_i)$ is the estimated probability of $Y=1$ for instance $\mathbf{x}_i$ and $p(\mathbf{x}_i)$ is the true probability of $Y=1$ for instance $\mathbf{x}_i$. $p(\mathbf{x}_i)$ is designated as one if the class label of $\mathbf{x}_i$ is one; otherwise, it is designated as zero. The Brier Score can be thought of as a measure of the calibration of a set of probabilistic predictions. It uses the estimated probability to provide an appropriate penalty. If the estimation technique predicts $Y=1$ for instance $\mathbf{x}_i$ with a high probability when in fact for $\mathbf{x}_i$, $Y=0$, the penalty will be higher than if it had predicted $Y=1$ with a low probability. Thus, the lower the Brier Score, the lower the penalty assessed for the estimation. In general, previous studies used Brier Score to evaluate calibration quality because the calibration method is regarded as an estimation technique.

Brier Score is decomposed in terms of Calibration Loss and Refinement Loss (Murphy, 1973). When the dataset is segmented into M subsets each of which shares an identical calibrated probability, Brier Score can be decomposed into these two additive components, as defined in Eq. (2.9):

$$\begin{aligned} &\text{Brier Score} = \text{Calibration Loss} + \text{Refinement Loss} \\ &= \tfrac{1}{N}\textstyle\sum_{k=1}^{M} n_k(\hat{p}(b_k) - \bar{p}(b_k))^2 + \tfrac{1}{N}\sum_{k=1}^{M} n_k(\bar{p}(b_k)(1-\bar{p}(b_k)) \end{aligned} \tag{2.9}$$

where $\hat{p}(b_k)$ is the estimated probability of $Y=1$ via the calibration method for the instances belonging to a subset b_k, and $\bar{p}(b_k)$ is the mean of the true

probability of $Y = 1$ for instances belonging to b_k. $\bar{p}(b_k)$ is directly obtained as a fraction of the positive data for each subset. The Calibration Loss term measures how close the estimated probabilities are to the true probabilities, given that estimation (Flach & Matsubara, 2007). Thus, Calibration Loss can itself be used as a measure of calibrated quality. The other term, Refinement Loss, is an integration of resolution and uncertainty (Murphy, 1972). This decomposition makes Brier Score an aggregate performance measure for binary-class data.

2.2.3 ROC curve

There are four possible performance measures for the evaluation of classification algorithms. If an instance is positive and he is classified as positive, he is counted as a True Positive (TP); if he is classified as negative, he is counted as a False Negative (FN). If the instance is negative and he is classified as negative, he is counted as a True Negative (TN); if he is classified as positive, he is counted as a False Positive (FP). Table 2.1 shows a confusion matrix and the equations of these common performance measures that can be calculated from it. The numbers along the main diagonal represent the correct decisions, and the numbers of the anti-diagonal represent the errors. For total positive instances, TP rate (TPR) means the rate of instances correctly classified and FN rate means the rate of instances incorrectly classified. For total negative instances, TN rate means the rate of instances correctly classified and FP rate (FPR) means the rate of instances incorrectly classified.

The ROC curve plots the TPR on the y-axis against the FPR on the x-axis (Fawcett, 2006). ROC curves were initially employed in signal detection theory

Table 2.1: Confusion matrix

		Actual class	
		P	N
Predicted class	$\hat{P}$	True Positive (TP)	False Positive (FP)
	$\hat{N}$	False Negative (FN)	True Negative (TN)

$$\text{TP rate} = \frac{TP}{P} \quad \text{FP rate} = \frac{FP}{N}$$

$$\text{FN rate} = \frac{FN}{P} \quad \text{TN rate} = \frac{TN}{N}$$

to depict the trade-off between TPRs and FPRs of classification algorithms (Egan, 1975; Swets et al., 2000). The use of ROC curves was extended to evaluate diagnostic information and make medical decisions (Pepe, 2003; Ren et al., 2004; Yang & Carlin, 2000). Recent years have seen an increase in the use of ROC curves in machine learning, in part due to the realization that simple classification accuracy is often poor at measuring performance (Bandos et al., 2007; Provost et al., 1998; P.-N. Tan et al., 2006).

Generally, ROC curves are used to evaluate the performance of classification algorithms and to check which algorithm is the most suitable. Some classification algorithms, such as NB, QDA, LDA, LR, and SVM, naturally yield probabilistic values when applied on a data and represent the degree to which an instance is a member of a class. We can calculate the score, the numeric value used directly for class prediction, by combining the function values. In the case of NB, QDA, and LDA that provide two kinds of function values, $p(\text{positive}/\mathbf{x})$ and $p(\text{negative}/\mathbf{x})$, the score, $p(\mathbf{x}{=}\text{positive})$, is calculated as $p(\text{positive}/\mathbf{x})$ divided by $p(\text{negative}/\mathbf{x})$, and ranges from 0 to ∞. Whereas LR and SVM algorithms yield a single function that computes the degree to which

an instance belongs to a class, and here, the functional value is directly used as the score. In LR, the score ranges from 0 to 1, while in SVM, the score does not have a limited range, but practically, its value is close to [-1, 1]. Classification algorithms can be used with a threshold to produce a discrete classifier: if the classifier score is above the threshold, the classifier produces a positive value and a negative otherwise. The ROC curve shows the changes in the TPR and FPR depending on the threshold used for classification. Each threshold value produces a different point in the ROC space. A threshold of ∞ classifies no instance into the maladjusted class and produces the point (0, 0). As the threshold is further reduced, it classifies more instances into the maladjusted class, and the curve climbs upwards to the right, ending at point (1, 1).

To demonstrate the generation of the ROC curve, we generate toy example of 60 instances as shown in Figure 2.1. Toy example has a much larger number of the negative than the positive. The instances are sorted by descending order of their score values. The ROC curve is created by varying threshold value. If the instance would be classified to the positive if its score is greater than the threshold. Figure 2.1 shows an ROC curve on the toy example, and each point is labelled by its ID number. Any ROC curve generated from a finite set of instances is actually a step function, which approaches a true curve as the numbers of the instances approach infinity. A threshold which is greater than the maximum value (99.05) produces the point (0, 0). As the threshold is further reduced, the curve climbs up and to the right, ending up at the point (1, 1) with a threshold less than 0.05. Smaller thresholds classify more instances into the positive class. On the ROC curve, a single instance matches with a single point

(FPR, TPR) when assuming that his score is used as the threshold. When a threshold is large, a few positive instances are classified into the positive. Therefore, the instances having large scores are matched with the ROC point where both TPR and FPR are low. On the whole, when a threshold is small, many instances are classified into the positive class. Therefore, the instances having small scores are matched with the ROC points where both TPR and FPR are large. We can identify that a threshold is directly and discretely matched with the score of individual instance in a finite dataset.

ROC points in a certain position are important to note. The lower left point (0, 0) represents the algorithm strategy of never issuing a positive classification; such a classification algorithm commits no false positive errors but also gains no true positives. The upper right point (1, 1) represents the opposite strategy, of unconditionally issuing positive classifications. The point (0, 1) represents perfect classification. Among the classification algorithms, there is no ROC point adjacent to point (0, 1). Generally, classification algorithm has the smaller FPR (in other words, the larger TN rate) in returns of the smaller TPR. ROC curves have an attractive property: they are insensitive to class skews. If the proportion of positive to negative instances changes in a test set, the ROC curves will not change. To see why this is so, consider the confusion matrix. We can note that the class distribution (the proportion of positive to negative instances) is the relationship of the left (positive) column to the right (negative) column. Any performance measure that uses values from both columns will be inherently sensitive to class skews. ROC curves are based upon TPR and FPR, in which each dimension is a strict columnar ratio, and so do not depend on

Ex.	Class	Score	Ex.	Class	Score	Ex.	Class	Score	Ex.	Class	Score
1	p	99.05	16	p	60.57	31	n	0.80	46	p	0.29
2	p	97.51	17	n	55.71	32	p	0.75	47	n	0.27
3	n	93.65	18	p	54.26	33	n	0.72	48	n	0.25
4	p	92.14	19	n	54.22	34	n	0.65	49	n	0.23
5	p	89.53	20	p	50.11	35	n	0.63	50	n	0.22
6	n	87.32	21	n	48.00	36	p	0.60	51	n	0.20
7	p	86.87	22	p	33.62	37	n	0.56	52	p	0.19
8	n	85.63	23	n	24.51	38	n	0.52	53	n	0.18
9	p	84.65	24	n	14.40	39	n	0.50	54	n	0.17
10	p	82.66	25	p	3.20	40	n	0.48	55	n	0.15
11	n	81.51	26	n	1.00	41	p	0.46	56	n	0.12
12	p	80.95	27	n	0.98	42	n	0.44	57	n	0.09
13	n	77.83	28	p	0.96	43	n	0.41	58	n	0.08
14	p	70.89	29	n	0.84	44	n	0.38	59	n	0.07
15	n	66.88	30	n	0.82	45	n	0.30	60	n	0.05

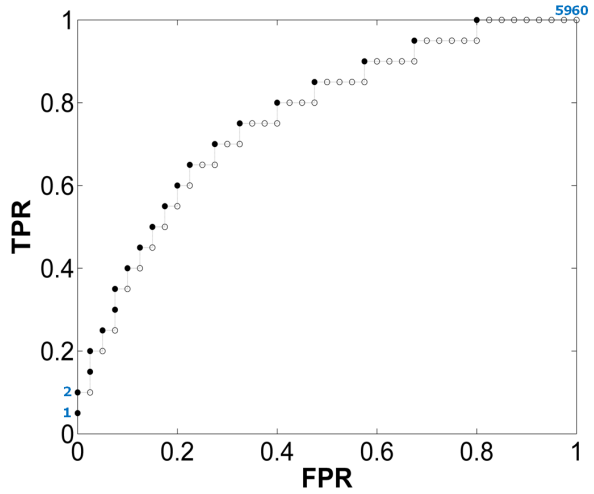


Figure 2.1: Toy example of ROC curve generation from a finite set of instances

class distributions.

2.3 ROC Binning

The calibration methods proposed in previous studies do not consider the change in the prevalence of the positive class for a dataset. Thus, their qual-

ity is likely to be poor when the prevalence of the positive class in the test set differs from that in the training set. When prevalence is considered, the calibration performance can be improved to become robust to changing class skews. Figure 2.2 exemplifies the effect of the prevalence of the positive class on the calibrated probability obtained by calculating the positive fraction for an instance subset S located within specific score values. The calibrated probability obtained from the training set will underestimate the true proportion of the positives in subset S of a test set when the prevalence of the positive class in the test set is greater than that in the training set. Conversely, it will overestimate the true proportion of positives in subset S for a test set in which the prevalence of the positive class is less than that in the training set. Previous studies that dealt with calibrated probability did not seriously consider the difference in class prevalence between the training and test sets.

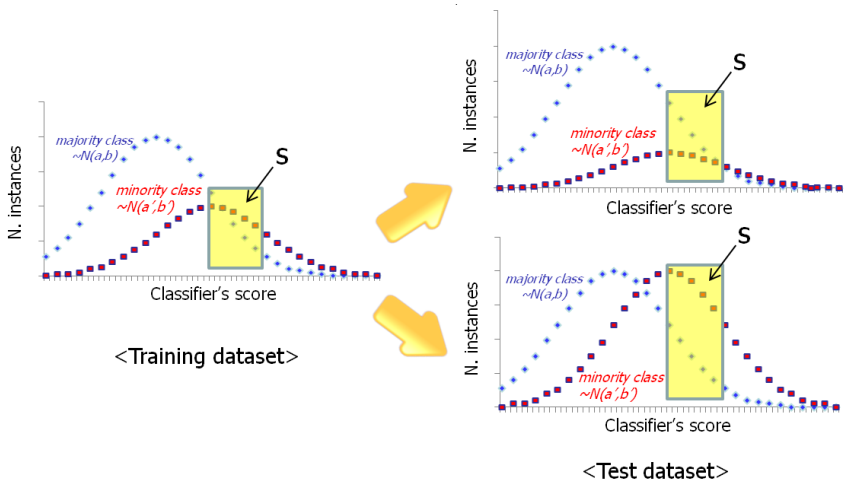


Figure 2.2: Influence of change in prevalence on calibrated probability

This study proposes a novel method, called ROC Binning, to obtain calibrated probability. The proposed method is effective in situations where the prevalence of the positive class changes over time or differs from that of the training set. It employs ROC curves to construct bins and obtains a calibrated probability using the ratio of the positives to the contents of the assigned bin. The main idea is to present a calibration method that is robust to changes in the class prevalence of a dataset. We utilize ROC curves because they are insensitive to class skews. That is, when the proportion of the positives to the negatives changes in a test set, ROC curves do not change (Fawcett, 2006). ROC graphs are based on TPR and FPR, which are respectively measured within an exclusive class value, and thus remain unaffected by class prevalence. The proposed ROC Binning method provides accurate calibrated probabilities by differentiating the distribution nature within a class value from the influence of class prevalence.

An ROC curve generated from a finite set of instances is actually represented as a step function, although it approaches a true curve as the number of instances approaches infinity. Each instance, $\mathbf{x}_i$, corresponds to an ROC point (tpr, fpr_i) where the score of instance $\mathbf{x}_i$ is the threshold. Further, the perimeter of the ROC curve, $\sum(\Delta TPR + \Delta FPR)$, is equal to two. The proposed ROC Binning method divides instances into M bins based on $\Delta TPR + \Delta FPR$ using Equal Width Interval Discretization (López et al., 2012) as defined in Eq. (2.10).

$$Equal\ Width\ Interval = \frac{maximum\ value - minimum\ value}{number\ of\ groups} \tag{2.10}$$

Because $(TPR + FPR)$ ranges from zero to two, the proposed ROC Binning method uses 2/(number of bins) as the interval criteria to make bins and estimates the calibrated probability using the positive fraction in each bin, b_k . The differential of $(TPR + FPR)$ between adjacent bins b_{k-1} and b_k, termed $\Delta(TPR+FPR)_k$, is always equal to 2/(number of bins) for every bin. However, ΔTPR_k and ΔFPR_k vary by b_k, respectively. Figure 2.3 demonstrates the respective changes in ΔTPR_k and ΔFPR_k on an ROC curve. The filled circles and the empty circles represent the ROC points corresponding to the positives and the negatives, respectively. A cross represents the boundary between bins where the number of bins is 10. ΔTPR_k has a larger value near the ROC point (0, 0) and decreases as it gets closer to (1, 1), whereas ΔFPR_k has a smaller value near the ROC point (0, 0) and increases as it gets closer to (1, 1). Thus, the positive proportion decreases as k increases.

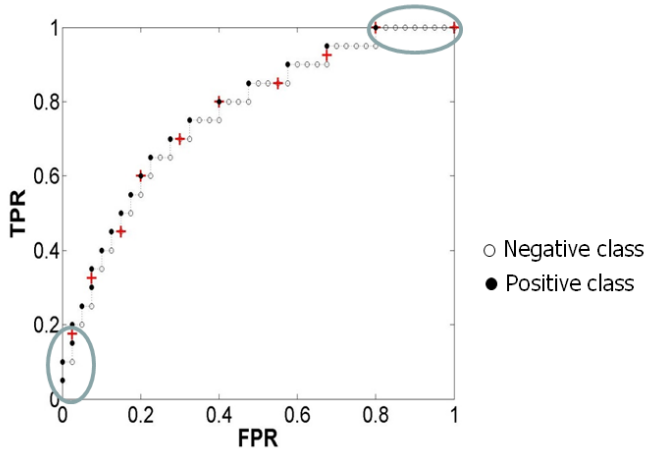


Figure 2.3: Changes of ΔTPR, ΔFPR, and positive proportion on the ROC curve

The procedure followed by the proposed ROC Binning method is shown in Figure 2.4. In the training phase, a classifier is constructed and the scores of the training instances calculated. Training instances are divided into M bins formed by Equal Interval of $(TPR + FPR)$. Bins b_1 to b_{M-1} have half-closed intervals containing only minimum points, and the last bin, b_M, has a closed interval.

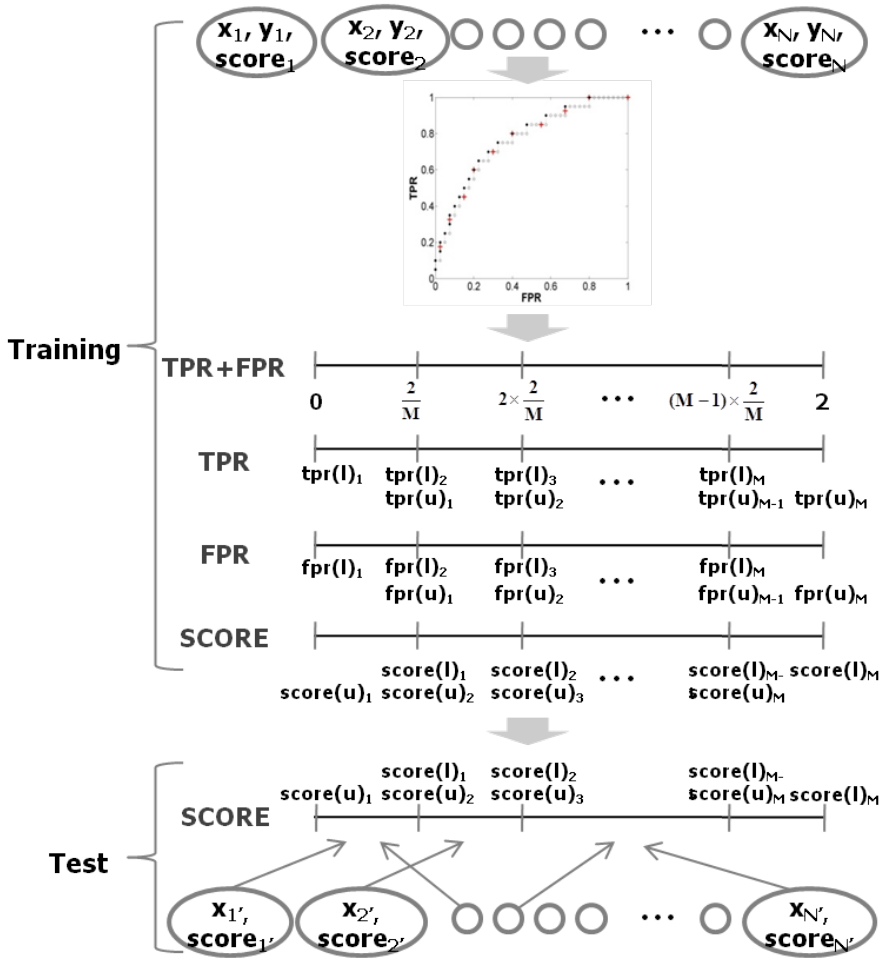


Figure 2.4: Procedure used by ROC Binning

For each b_k, the TPR range, specifically, $[b\_tpr(l)_k,\ b\_tpr(u)_k)$; the FPR range, specifically, $[b\_fpr(l)_k,\ b\_tpr(u)_k)$;, and the score range, specifically, $(b\_score(l)_k,\ b\_score(u)_k]$, can be found. The calibrated probability, or positives proportion, for the training instances that have scores within $(b\_score(l)_k,\ b\_score(u)_k]$, $\hat{p}_{RB}(\mathbf{x}_{b_k})$, satisfy Eq. (2.11):

$$\begin{aligned}\hat{p}_{RB}(x_{b_k}) &= \frac{\text{N. positive instances in scores of } (b\_score(l)_k,\ b\_score(u)_k]}{\text{N. instances in scores of } (b\_score(l)_k,\ b\_score(u)_k]} \\ &= \frac{(b\_tpr(u)_k - b\_tpr(l)_k) \times p(\text{positive})}{(b\_tpr(u)_k - b\_tpr(l)_k) \times p(\text{positive}) + (\text{b\_fpr(h)}_\text{k} - \text{b\_fpr(l)}_\text{k}) \times (1 - \text{p(positive)})}\end{aligned} \tag{2.11}$$

where $p(positive)$ represents the positive class prevalence.

In the test phase, ROC Binning assigns the test instances to predefined bins by adopting the classifier's score. Then, it estimates the calibrated probability of being positive for instances belonging to a b_k using Eq. (2.11). Here, the $p(positive)$ of the test set is different from that of the training set. The computation algorithm used by ROC Binning is outlined in Figure 2.5.

Although test and training sets have different positive class prevalence, each bin of the test set is expected to have the same TPR and FPR as that of the corresponding bin of the training set because these measures are insensitive to class skews. ROC Binning distinguishes the distribution nature within a class value and the influence of the prevalence using ROC curves and reflects the difference in positive class prevalence between the test and training sets. The proposed method, ROC Binning, can be a very useful calibration method to cope with changing prevalence. This study assumes that the positive class prevalence is known. According to previous studies (Barranquero et al., 2013; Forman, 2008), this enables us to estimate the prevalence relatively well, even

with an imperfect classifier.

Inputs: $\{(\mathbf{x}_i, y_i), i = 1, ..., N\}$, the training set; $\{\mathbf{x}_j, j = 1, ..., N'\}$, the test patterns; $p(positive)$, the positive class prevalence of test set; M, number of bins

Output: $(\{\hat{p}_{RB}(\mathbf{x}_{b_k}), k = 1, ..., M\}$, the calibrated probability for patterns in each bin using ROC Binning

// Training phase:

Step 1: Construct a classifier from the training dataset

$Classifier \leftarrow$ a leaning model that fits the relationship between $\mathbf{x}$ and y

Step 2: Calculate the training pattern's score by employing the Classifier

For each pattern $i = 1 : N$

$score_i \leftarrow (f\_Scoring \otimes Classifier)(\mathbf{x}_i)$

Step 3: Generate ROC points from a finite training set

For each pattern $i = 1 : N$

$(tpr_i, fpr_i, threshold_i) \leftarrow$ ROC point when $score_i$ is the threshold

Step 4: Divide TPR+FPR into M bins with Equal Width Interval

$\{[0, (tpr+fpr)_1), ..., [(tpr+fpr)_{M-1}, (tpr+fpr)_M]\} \leftarrow \{[0, \frac{2}{M}), ...[(M-1)\times\frac{2}{M}, M\times\frac{2}{M}]\}$

Step 5: Find the parameters (b_TPR, b_FPR, b_SCORE) for ROC Binning

For each bin $k = 1 : M$

$(b\_tpr(l)_k, b\_tpr(u)_k) \leftarrow \{tpr_i : \ tpr_i + fpr_i \in [(tpr + fpr)_{k-1}, (tpr + fpr)_k)\}$

$(b\_fpr(l)_k, b\_fpr(u)_k) \leftarrow \{fpr_i : \ tpr_i + fpr_i \in [(tpr + fpr)_{k-1}, (tpr + fpr)_k)\}$

$(b\_score(l)_k, b\_score(u)_k) \leftarrow \{\text{threshold}_\text{i} : \ tpr_i + fpr_i \in [(tpr + fpr)_{k-1}, (tpr + fpr)_k)\}$

// Test phase:

Step 6: Calculate the test pattern's score by employing the Classifier

For each pattern $j = 1 : N'$

$score_j \leftarrow (f\_Scoring \otimes Classifier)(\mathbf{x}_j)$

Step 7: Estimate the calibrated probability for patterns belonging to a bin, k

For each bin $k = 1 : M$

$\mathbf{x}_{b_k} \leftarrow \{\text{x}_j : \ score_j \in (b\_score(l)_k, b\_score(u)_k)\}$

$$\hat{p}_{RB}(\mathbf{x}_{b_k}) = \frac{(b\_tpr(u)_k - b\_tpr(l)_k) \times p(\text{positive})}{(b\_tpr(u)_k - b\_tpr(l)_k) \times p(\text{positive}) + (\text{b\_fpr(h)}_\text{k} - \text{b\_fpr(l)}_\text{k}) \times (1 - \text{p(positive)})}$$

Figure 2.5: Computation algorithm utilized by ROC Binning

2.4 Performance on benchmark datasets

2.4.1 Experiment settings

We conducted several experiments designed to verify the efficacy of the proposed ROC Binning method. Given a training set in which the prevalence of the positive class is noticeably different from that in the test set, this method should effectively obtain the calibrated probability of the test set. Therefore, the verification requires performance evaluation over a broad spectrum of test sets with different positive class prevalence, as opposed to using just a single test set. The experimental methodology used by Forman (Forman, 2008) and Barranquero (Barranquero et al., 2013) was used to vary the test conditions.

2.4.1.1 Experiment methodology

Benchmark datasets with known class labels were selected for performance measurement and comparison. The stratified ten-fold cross-validation procedure was adopted to take into account the specific purpose for our study. In this procedure, the original positive class prevalence of a dataset is preserved in training iterations and varied in test iterations. A dataset is partitioned in order that the percentage of instances that have a single class value in each fold is similar to the percentage in another fold. Once a classification model is learned with nine of the folds, the remaining one is used to generate 11 different random test sets with specific positive proportions ranging from 0% to 100%, in steps of 10%, following the experimental methodology used by (Forman, 2008) and Barranquero (Barranquero et al., 2013). We added one more test condi-

tion whose positive proportion is identical to that of the training set, because it is also meaningful to assess the calibration performance in the conventional experimental scenario and to compare that with the aforementioned 11 testing conditions. This variation in the total of 12 testing conditions is appropriate for evaluating the robustness of calibration methods under conditions of changing positive class prevalence.

2.4.1.2 Datasets

In order to conduct a fair comparison of ROC Binning with conventional calibration methods, we searched for a collection of datasets retaining numerical (real, integer) attributes and categorical dependant variables from the UCI Machine Learning Repository (Lichman, 2013). We used 26 datasets with no values missing. Detailed descriptions of each dataset are given in Table 2.2. The datasets contained varying numbers of attributes, class values, and patterns. The number of attributes ranged from three (Haberman's Survival) to166 (Musk); the number of class values ranged from two (Haberman's Survival) to 11 (Vowel Recognition); and the number of patterns ranged from 106 (Breast Tissue) to 64,448 (Handwritten Digits). Twenty-six datasets were recomposed into 87 binary-class datasets using a one-vs.-rest approach (as summarized in Table 2.3), in order to evaluate the calibration method efficiently. For innately binary-class datasets, the minority class and majority class of the original dataset was designated as positive class and negative class, respectively. Note that the proportion of positive instances varied from 4.3% to 77.8%. This fact offered the possibility of evaluating the calibration methods over significantly different degrees of class

imbalance. We selected binary-class datasets, in which the number of minority class instances is equal to or more than 10, to avoid tests conditions that result in no positive instance being available, because ten-fold cross validation was used in the experiment.

Table 2.2: Summary of datasets

Dataset	Attributes	Classes	Instances	% fewest class
Haberman's Survival	3	2	306	26.5
Iris	4	3	150	33.3
Blood Transfusion Center	4	2	748	23.8
Balance Scale	4	3	625	7.8
Liver Disorders	6	2	345	42.0
Seeds	7	3	219	11.9
Ecoli	7	8	336	5.8
Pima Indians Diabetes	8	2	768	34.9
Breast Tissue	9	6	106	13.2
Breast Cancer Wisconsin	9	2	683	35.0
Tic-Tac-Toe Endgame	9	2	958	34.7
Contraceptive Method Choice	9	3	1,473	22.6
Glass Identification	10	6	214	6.1
Vowel Recognition	10	11	990	9.1
Wine	13	3	180	27.2
Statlog (Vehicle Silhouettes)	18	4	846	23.5
Image Segmentation	19	7	2,310	14.3
Cardiotocography	22	3	2,126	8.3
Parkinsons	22	2	195	24.6
Ionosphere	34	2	351	35.9
SPECTF heart	44	2	267	20.6
Sonar	60	2	208	46.6
Optical Recognition of Digits	64	10	2,903	9.5
Hill-Valley	100	2	1,212	49.5
Urban Land Cover	147	9	675	13.8
Musk	166	2	475	43.6

Estimating calibrated probability is a more significant issue especially in class imbalance and class overlap problems in which direct application of classification model predictions may result in substantial errors. Therefore, we checked the properties related to them, the proportion of the positive class, and the overlap degree for two classes, as shown in Table 2.3. We can also see the distributions of the first and second PCs of the benchmark datasets in Appendix A. Class imbalance can be simply identified by counting the number of positives(Pos.) and the number of negatives. However, identifying the class overlap degree requires a more in-depth approach. Therefore, we computed the volume of the overlap region (F2) (Sánchez et al., 2007) for two classes, which is the product of normalized lengths of overlapping ranges for all attributes:

$$F2 = \prod_{i} \frac{MIN(\max(\mathbf{x}_i, c_1), \max(\mathbf{x}_i, c_0) - MAX(\min(\mathbf{x}_i, c_1), \min(\mathbf{x}_i, c_0)}{MAX(\max(\mathbf{x}_i, c_1), \max(\mathbf{x}_i, c_0) - MIN(\min(\mathbf{x}_i, c_1), \min(\mathbf{x}_i, c_0)} \qquad (2.12)$$

where i=1, . . . , d for a d-dimensional attribute set, and c_1 and c_0 are, respectively, the class labels of the positive and the negative.

In Eq. (2.12), if there is any attribute with numerator equal to or less than zero, then the value of F2 will be zero. We computed F2 not only for all attributes but also for the first principal component, as shown in Table 2.3. Note that the F2 varied from 0% to 100%. This fact offered the possibility of evaluating the calibration methods over significantly different degrees of class overlap.

Table 2.3: Summary of recomposed binary-class datasets

Datasets	Positive class	Identifier	% Pos.	F2(%)	1st PC's F2(%)
Haberman's Survival	survived	Haberman	26.5	71.8	73.6
Iris	setosa	Iris.1	33.3	0.0	0.0
	versicolour	Iris.2	33.3	3.5	18.1
	virginica	Iris.3	33.3	0.7	13.3
Blood Transfusion	donating blood	Blood	23.8	27.1	33.0
Balance Scale	left	Balance.1	7.8	100.0	100.0
	balanced	Balance.2	46.1	100.0	100.0
	right	Balance.3	46.1	100.0	100.0
Liver Disorders	positive	Liver	42.0	7.3	52.1
Seeds	kama	Seeds.1	33.3	3.1	31.1
	rosa	Seeds.2	33.3	0.0	11.9
	canadian	Seeds.3	33.3	0.1	18.4
Ecoli	cp	Ecoli.1	42.6	0.0	5.8
	im	Ecoli.2	22.9	0.0	18.4
	imU	Ecoli.3	10.4	0.0	19.8
	om	Ecoli.4	6.0	0.0	8.4
	pp	Ecoli.5	15.5	0.0	11.5
Pima Indians Diabetes	positive	Pima	34.9	25.2	75.7
Breast Tissue	carcinoma	Breast.1	19.8	0.0	7.5
	fibro-adenoma	Breast.2	14.2	0.0	2.0
	mastopathy	Breast.3	17.0	0	9.5
	glandular	Breast.4	15.1	0.0	4.2
	connective	Breast.5	13.2	0.0	6.7
	adipose	Breast.6	20.8	0	15.1
Breast Wisconsin	malignant	Wisconsin	35.0	21.7	27.8
Tic-Tac-Toe Endgame	positive	Tic-Tac-Toe	34.7	100.0	86.6
Contraceptive Method	no-use	Contraceptive.1	42.7	75	71.4
	left	Contraceptive.2	22.6	72.7	60.4
	short-term	Contraceptive.3	34.7	81.3	72.5
Glass	windows float	Glass.1	34.1	0.0	9.8
	windows non float	Glass.2	37.1	0.3	23.8
	vehicle windows	Glass.3	8.3	0.0	4.2
	tableware	Glass.4	6.3	0.0	40.9
	headlamps	Glass.5	14.1	0.0	34.4
Vowel Recognition	type 0	Vowel.1	9.1	1.2	48.6
	type 1	Vowel.2	9.1	0.3	27.0
	type 2	Vowel.3	9.1	0.0	4.6

Continued

Table 2.3 –continued from previous page

Datasets	Positive class	Identifier	% Pos.	F2(%)	1st PC's F2(%)
	type 3	Vowel.4	9.1	0.4	15.7
	type 4	Vowel.5	9.1	0.2	15.5
	type 5	Vowel.6	9.1	1.4	45.4
	type 6	Vowel.7	9.1	1.3	37.0
	type 7	Vowel.8	9.1	1.0	31.0
	type 8	Vowel.9	9.1	1.9	33.8
	type 9	Vowel.10	9.1	0.1	9.7
	type 10	Vowel.11	9.1	1.8	68.3
Wine	type 1	Wine.1	33.3	0.0	12.0
	type 2	Wine.2	39.4	0.1	9.2
	type 3	Wine.3	27.2	0.0	7.5
Statlog	opel	Statlog.1	23.5	0.0	53.2
	saab	Statlog.2	25.7	0.0	61.9
	bus	Statlog.3	25.8	0.3	45
	van	Statlog.4	25.1	0.1	66.6
Image Segmentation	image 1	Image.1	14.3	0.0	2.3
	image 2	Image.2	14.3	0.0	0.0
	image 3	Image.3	14.3	0.0	19.9
	image 4	Image.4	14.3	0.0	16.7
	image 5	Image.5	14.3	0.0	7.4
	image 6	Image.6	14.3	0.0	4.6
	image 7	Image.7	14.3	0.0	2.3
Cardiotocography	normal	Cardio.1	77.8	0.3	61.1
	suspect	Cardio.2	13.9	0.0	73.2
	pathologic	Cardio.3	8.3	0.0	38.8
Parkinsons	disease	Parkinsons	24.6	0.0	17.9
Ionosphere	bad	Ionosphere	35.9	0.0	58.9
SPECTF heart	abnormal	SPECTF	20.6	0.0	5.6
Sonar	rock	Sonar	46.6	0.0	76.1
Recognition of Digit	number 0	Digits.1	9.9	0.0	8.7
	number 1	Digits.2	10.2	0.0	21.4
	number 2	Digits.3	10.0	0.0	19.6
	number 3	Digits.4	10.0	0.0	15.9
	number 4	Digits.5	10.1	0.0	33.8
	number 5	Digits.6	10.1	0.0	20.8
	number 6	Digits.7	10.0	0.0	8.0
	number 7	Digits.8	10.1	0.0	26.2
	number 8	Digits.9	9.5	0.0	13.9
	number 9	Digits.10	10.1	0.0	34.8

Continued

Table 2.3 –continued from previous page

Datasets	Positive class	Identifier	% Pos.	F2(%)	1st PC's F2(%)
Hill-Valley	hill	Hill-Valley	49.5	0.0	47.5
Urban Land Cover	asphalt	Urban.1	8.7	0.0	21.6
	building	Urban.2	18.1	0.0	48.9
	car	Urban.3	5.3	0.0	40.2
	concrete	Urban.4	17.2	0.0	45.0
	grass	Urban.5	16.6	0.0	37.7
	pool	Urban.6	4.3	0.0	13.8
	shadow	Urban.7	9.0	0.0	20.2
	soil	Urban.8	5.0	0.0	23.5
	tree	Urban.9	15.7	0.0	31.4
Musk	musk	Musk	43.6	0.0	72.9

2.4.1.3 Classification Algorithms and Parameters

We used the well-known classification algorithms Naïve Bayes (NB), Bayesian Quadratic and Discriminant Analysis (QDA), and Linear Discriminant Analysis (LDA), Logistic Regression (LR), and Support Vector Machine (SVM) as base-learners for calibration. These classification algorithms naturally yield function values that represent the degree to which a pattern is a member of a class. We can calculate the *score*, the numeric value used directly for class prediction, by combining the function values. In the case of NB, QDA, and LDA, which provide two kinds of function values, $p(\text{positive}/\mathbf{x})$ and $p(\text{negative}/\mathbf{x})$, the score is calculated as $p(\text{positive}/\mathbf{x})$ divided by $p(\text{negative}/\mathbf{x})$, and ranges from zero to ∞. The LR and SVM algorithms yield a single function that computes the degree to which a pattern belongs to a class; here, its functional value is used directly as the score. In LR, the score ranges from zero to one, while in SVM, the score does not have a limited range, but practically, its value is close to [-1,

1]. SVM models have a cost parameter, C, that controls the trade-off between allowing training errors and forcing rigid margins. We selected C using ten-fold cross validation and grid search from candidates [0.01 0.1 0 1 10]. NB assumes the class conditional probability to be a Gaussian distribution, in common with LDA and QDA.

2.4.1.4 Calibration Methods and Comparison Measures

We empirically compared the proposed ROC Binning (RB) method to popular calibration methods: Histogram Binning (HB), Platt Scaling (PS), and Isotonic Regression (IR). Their implementation was performed off-line using the commercial software package MATLAB (*Matlab version 7.10.0*, 2010). The software for the proposed RB and HB was developed by us. The other calibration methods, PS and IR, were actualized using the source code developed in previous studies (*Isonotic Regression Software*, 2005; Lin et al., 2007).

In the proposed RB method, the calibrated probability depends on the number of alternative bins. In order to reduce the variance of the probability estimates, the number of bins, M, should not be overly large. In the experiments, we set M as 10. HB had the same number of bins as RB. In PS, the model parameters were estimated using the Newton algorithm. The PAV algorithm was employed to minimize mean square error in nonparametric IR.

This study measures Brier Score (BS), which is commonly used to evaluate calibration methods, and Calibration Loss (CL), which requires the subsets of data sharing the identical calibrated probability. RB, HB, and IR naturally yield step functions of the calibrated probability and thus have several subsets

of instances that discretely share the same calibrated probability. However, PS provides a continually increasing function of the calibrated probability and thus has the number of subsets being virtually identical to that of total instances.

2.4.2 Experiment results

We evaluated four calibration methods on 87 binary-class datasets by applying a stratified ten-fold cross-validation and varying the positive proportion of the test set at $\{p(\text{training}), 0, 10\%, \ldots, 90\%, 100\%\}$. This resulted in 870 training processes and 10440 tests for every combination (4 calibration methods $\times$ 5 classification algorithms). The execution was not analyzed because it is heavily dominated by learning a classification model and is only marginally affected by the post-processing of the calibration itself.

2.4.2.1 Overview analysis

BS was evaluated for four calibration methods with 12 test conditions using five classification algorithms as base-learners, respectively. Appendix B shows the distribution of BS according to test condition when NB was used for each dataset. RB exhibited the best performance across different positive prevalence for virtually every dataset. We averaged the BS from 12 repetitions (12 test conditions) for each binary-class dataset. When using NB, the averaged BS of each calibration method for each binary-class datasets is shown in Table 2.4. Most datasets resulted in similar characteristics: RB and IR exhibited the best and second-best performances for most datasets, respectively. The difference between the two ranged from -0.001 to 0.251. The other classification algorithms

had results that were equivalent to that of NB.

Table 2.4: BS for each binary-class dataset (e.g., NB)

Identifier	HB	PS	IR	RB
Haberman	0.274	0.279	0.278	0.156
Iris.1	0.046	0.000	0.010	0.010
Iris.2	0.058	0.068	0.066	0.035
Iris.3	0.052	0.057	0.050	0.040
Blood	0.273	0.285	0.271	0.136
Balance.1	0.395	0.403	0.408	0.157
Balance.2	0.093	0.093	0.095	0.071
Balance.3	0.102	0.101	0.107	0.081
Liver	0.240	0.251	0.239	0.151
Seeds.1	0.085	0.095	0.089	0.063
Seeds.2	0.038	0.032	0.037	0.024
Seeds.3	0.045	0.054	0.049	0.037
Ecoli.1	0.041	0.130	0.034	0.022
Ecoli.2	0.131	0.150	0.123	0.066
Ecoli.3	0.186	0.242	0.182	0.079
Ecoli.4	0.130	0.201	0.138	0.053
Ecoli.5	0.094	0.283	0.091	0.054
Pima	0.192	0.199	0.192	0.117
Breast.1	0.101	0.100	0.086	0.034
Breast.2	0.286	0.256	0.287	0.115
Breast.3	0.304	0.282	0.287	0.125
Breast.4	0.216	0.278	0.222	0.111
Breast.5	0.163	0.200	0.157	0.059
Breast.6	0.089	0.059	0.090	0.047
Wisconsin	0.041	0.038	0.034	0.027
Tic-Tac-Toe	0.223	0.230	0.226	0.135
Contraceptive.1	0.234	0.234	0.234	0.147
Contraceptive.2	0.275	0.275	0.276	0.138
Contraceptive.3	0.257	0.256	0.255	0.151
Glass.1	0.079	0.116	0.077	0.047
Glass.2	0.116	0.101	0.110	0.076

Continued

Table 2.4 –continued from previous page

Identifier	HB	PS	IR	RB
Glass.3	0.126	0.185	0.114	0.076
Glass.4	0.222	0.279	0.165	0.076
Glass.5	0.073	0.071	0.058	0.036
Vowel.1	0.261	0.300	0.257	0.076
Vowel.2	0.145	0.152	0.130	0.041
Vowel.3	0.136	0.135	0.125	0.042
Vowel.4	0.219	0.212	0.209	0.080
Vowel.5	0.254	0.260	0.244	0.089
Vowel.6	0.225	0.245	0.217	0.080
Vowel.7	0.109	0.109	0.103	0.026
Vowel.8	0.247	0.264	0.251	0.079
Vowel.9	0.147	0.172	0.137	0.052
Vowel.10	0.237	0.240	0.212	0.074
Vowel.11	0.078	0.087	0.084	0.028
Wine.1	0.035	0.039	0.036	0.020
Wine.2	0.031	0.027	0.031	0.023
Wine.3	0.021	0.019	0.025	0.014
Statlog.1	0.214	0.247	0.215	0.108
Statlog.2	0.255	0.262	0.251	0.133
Statlog.3	0.188	0.204	0.183	0.097
Statlog.4	0.252	0.263	0.251	0.132
Image.1	0.083	0.163	0.063	0.032
Image.2	0.055	0.013	0.005	0.004
Image.3	0.142	0.134	0.130	0.065
Image.4	0.181	0.211	0.177	0.077
Image.5	0.235	0.250	0.235	0.097
Image.6	0.063	0.047	0.033	0.023
Image.7	0.058	0.015	0.007	0.008
Cardiotocography.1	0.121	0.142	0.114	0.062
Cardiotocography.2	0.145	0.228	0.140	0.063
Cardiotocography.3	0.150	0.190	0.180	0.066
Parkinsons	0.198	0.240	0.193	0.103
Ionosphere	0.114	0.113	0.102	0.072
SPECTF heart	0.223	0.290	0.213	0.110
Sonar	0.233	0.220	0.235	0.163

Continued

Table 2.4 –continued from previous page

Identifier	HB	PS	IR	RB
Digits.1	0.013	0.223	0.014	0.010
Digits.2	0.111	0.342	0.110	0.047
Digits.3	0.073	0.286	0.075	0.040
Digits.4	0.098	0.339	0.100	0.048
Digits.5	0.063	0.104	0.062	0.030
Digits.6	0.052	0.348	0.054	0.038
Digits.7	0.029	0.221	0.031	0.025
Digits.8	0.073	0.281	0.071	0.031
Digits.9	0.169	0.379	0.168	0.070
Digits.10	0.144	0.337	0.141	0.059
Hill-Valley	0.251	0.250	0.250	0.160
Urban.1	0.103	0.390	0.111	0.054
Urban.2	0.145	0.334	0.152	0.078
Urban.3	0.225	0.420	0.155	0.076
Urban.4	0.127	0.336	0.136	0.068
Urban.5	0.148	0.342	0.153	0.075
Urban.6	0.241	0.431	0.127	0.054
Urban.7	0.126	0.387	0.144	0.054
Urban.8	0.270	0.423	0.242	0.093
Urban.9	0.114	0.347	0.113	0.050
Musk	0.258	0.254	0.271	0.132

Table 2.5 summarizes the BS obtained for each calibration method over all the binary-class datasets. First, the average of BS was calculated from all datasets for each calibration method. Then, for each method, we counted the cases in which it had the best BS among all calibration methods across all datasets. Among all the compared calibration methods, the proposed RB method was the best across the datasets. The superiority of RB was decisive in terms of the abovementioned average and number of datasets on which its

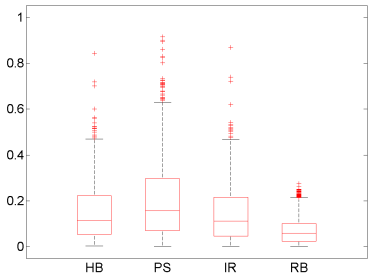
performance was the best, for every classification algorithm. IR was second-best among all the classification algorithms. The performance of PS of the sigmoid function using LR was slightly different from the performance with IR. RB outperformed all the other methods on 61∼85 out of the total of 87 datasets for every classification algorithm.

Table 2.5: Summary of BS for all the test conditions

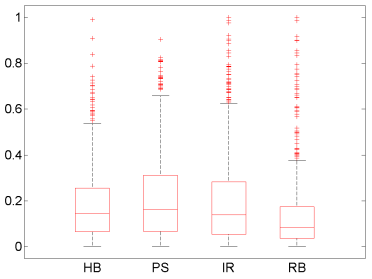
Classfier	Measure	HB	PS	IR	RB
NB	Average	0.151	0.209	0.145	0.071
	N. best datasets	0	1	1	85
QDA	Average	0.188	0.222	0.202	0.132
	N. best datasets	7	7	3	70
LDA	Average	0.186	0.236	0.190	0.085
	N. best datasets	0	0	3	84
LR	Average	0.155	0.156	0.154	0.090
	N. best datasets	9	10	0	68
SVM	Average	0.136	0.338	0.130	0.087
	N. best datasets	10	0	16	61

Figure 2.6 graphically depicts the BS results by box-plot for all 12 test conditions in all 87 datasets. The range of BS can be observed for every method. The results for all the classification algorithms are virtually equivalent. Considering the main elements of the box-plot, RB is the most compact method in terms of the inter-quartile range. For example, in NB, RB has its first quintile, its median, and its third quintile around 0.025, 0.059, and 0.101, respectively. Each of them is the lowest value of all the calibration methods. Thus, RB gives

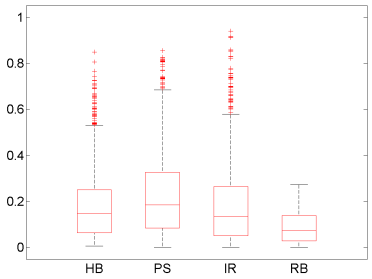
the most competitive calibrated probability across al1 12 test conditions, with prevalence varying from 0% to 100%.



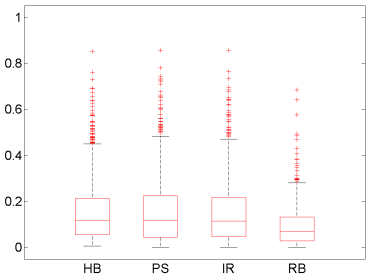
(a) NB



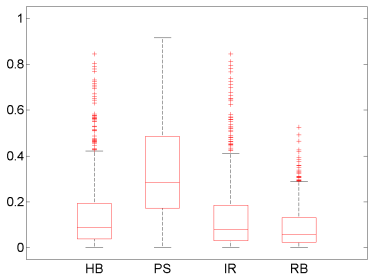
(b) QDA



(c) LDA



(d) LR



(e) SVM

Figure 2.6: Box-plots of BS for all the test conditions

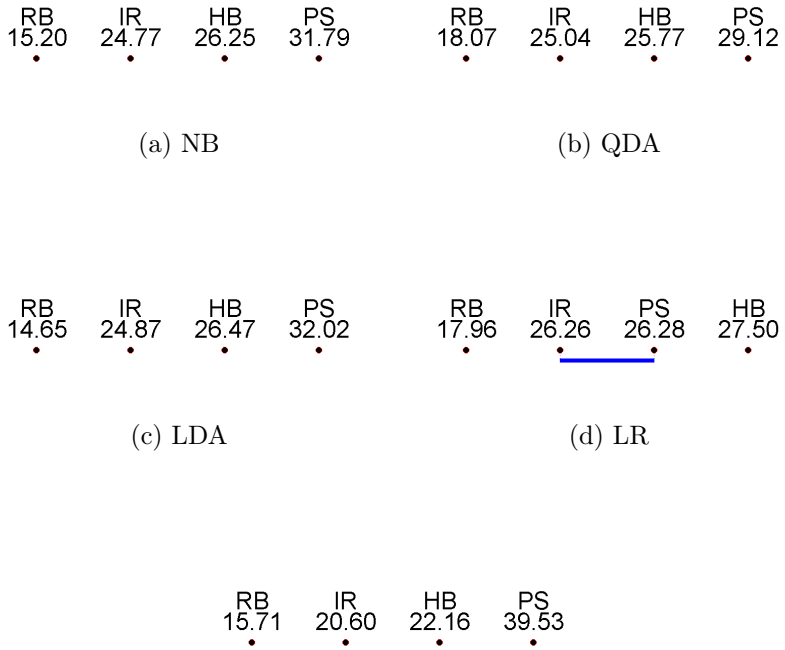
We also performed a two-step statistical test, in accordance with Demšar's statement (Demšar, 2006) that the Friedman test (M. Friedman, 1940) with corresponding post-hoc tests is recommended for comparison of performance with multiple algorithms over multiple datasets. The first step was a Friedman test of the null hypothesis that all calibration methods perform equally in terms of BS rank. When this hypothesis was rejected, a Nemenyi post-hoc test was conducted for pair-wise comparison of the multiple calibration methods. Both tests are based on the rank of every calibration method. First, they aggregate the ranks of the calibration methods obtained per test condition and then compute the average rank for each dataset. Four calibration methods were compared over 87 binary-class datasets, with 12 repetitions.

Friedman's null hypothesis was rejected at the 5% significance level for each classification algorithm. The results of the post-hoc Nemenyi test (Nemenyi, 1963), in which the calibration methods are sorted by average rank in ascending order, are shown in Figure 2.7. In the Nemenyi test, the null hypothesis (no performance difference between algorithms) was rejected when the difference in the average ranks was greater than the critical distance (CD), with

$$CD = q_\alpha \sqrt{\frac{A(A+1)}{6B}} \tag{2.13}$$

where A and B are the number of models and datasets, respectively, and critical values q_α are based on the studentized range statistic divided by $\sqrt{2}$.
The CD for the Nemenyi test with 87 datasets and four methods was 0.503 at α=0.05. In Figure 2.7, the average rank is written next to the calibration method's name. There is no significant difference among the calibration meth-

ods joined by the horizontal lines. Thus, implies that RB performed significantly better than the other methods in terms of BS.



(a) NB

(b) QDA

(c) LDA

(d) LR

(e) SVM

Figure 2.7: Nemenyi post-hoc test for all the test conditions at α=0.05

Table 2.6 summarizes the analysis results of the CL of those calibration methods for all datasets across the different classification algorithms. As with BS, we analyzed the CL results in terms of the average and the method that performed the best on the most datasets. As was the case for BS, the performance of RB was superior to that of the other methods. In short, we obtained

results for BS and CS that were equivalent for comparison of the calibration methods in changing prevalence.

Table 2.6: Summary of CL for all the test conditions

Classfier	Measure	HB	PS	IR	RB
NB	Average	0.105	0.189	0.101	0.029
	N. best datasets	0	1	0	86
QDA	Average	0.128	0.188	0.137	0.075
	N. best datasets	8	5	7	67
LDA	Average	0.128	0.218	0.132	0.031
	N. best datasets	0	0	2	85
LR	Average	0.110	0.156	0.104	0.049
	N. best datasets	6	3	13	65
SVM	Average	0.099	0.338	0.084	0.049
	N. best datasets	3	0	24	60

2.4.2.2 Analysis of results per test prevalence

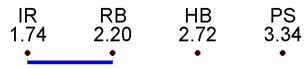
Although the foregoing account provides interesting evidence verifying the quality of RB, the effect of the positive class prevalence in the test set is not shown in detail. For example, only an aggregate overview of the distributions of BS is given, without any consideration given to the error distribution with regard to the different test prevalence. In this subsection, we analyze the performance of each calibration method with respect to changes in test prevalence. In Appendix B, the vertical gray line represents the positive class prevalence of the training set. The BS of the conventional calibration methods significantly in-

creased because the positive class prevalence is much different from that of the training set. In contrast, in RB, BS remains stable or even decreases. Table 2.7 summarizes the resulting BS per test condition over all the datasets. RB gives the best performance across 0–100% different test set prevalence conditions. The performance of RB is slightly lower than that of the IR method when the positive prevalence of the test set is identical to that of the training set.

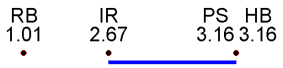
To compare the BS obtained by the four calibration methods per prevalence over 87 datasets a Friedman test with the corresponding post-hoc tests was conducted with no repetition. Friedman's null hypothesis was rejected at α=5% for every test prevalence. CD for the Nemenyi test with 87 datasets and the four methods was 0.503 at α=5%, which is identical in that respect to the aggregate prevalence analysis in subsection 5.2 because it is not dependent on repetition. The overall results of the Nemenyi test are shown in Figure 2.8. In cases where the positive prevalence of the test set was identical to that of the training set, although the performance of RB was slightly lower than that of IR, there was no statistically significant difference between the two. RB was the best for the other 11 test conditions, often with statistically significant superiority. Although the performance of IR is slightly better when the test condition is equal to that of the training, it did not dominate a large variety of test conditions. This fact leads us to conclude that RB is the most suitable method to deal with a variety of datasets and test conditions.

Table 2.7: BS per test condition (e.g., NB)

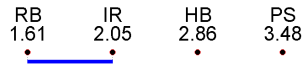
% positives	Measure	HB	PS	IR	RB
p(training)	Average	0.081	0.094	0.075	0.076
	N. best Brier Socre	12	10	42	23
0	Average	0.048	0.048	0.039	0
	N. best Brier Socre	0	0	1	87
10	Average	0.070	0.081	0.061	0.049
	N. best Brier Socre	4	3	29	51
20	Average	0.092	0.116	0.084	0.078
	N. best Brier Socre	8	5	16	58
30	Average	0.113	0.150	0.106	0.096
	N. best Brier Socre	9	7	16	55
40	Average	0.133	0.181	0.126	0.105
	N. best Brier Socre	8	9	8	62
50	Average	0.157	0.217	0.150	0.109
	N. best Brier Socre	4	9	4	70
60	Average	0.179	0.252	0.173	0.107
	N. best Brier Socre	2	6	2	77
70	Average	0.202	0.289	0.196	0.096
	N. best Brier Socre	1	4	2	80
80	Average	0.230	0.329	0.224	0.075
	N. best Brier Socre	0	2	1	84
90	Average	0.247	0.359	0.243	0.050
	N. best Brier Socre	0	2	1	84
100	Average	0.265	0.387	0.261	0.012
	N. best Brier Socre	0	2	0	85



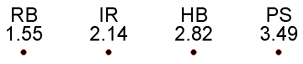
(a) p(training)



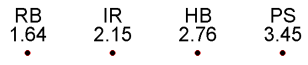
(b) 0%



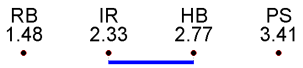
(c) 10%



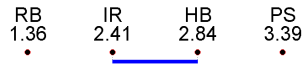
(d) 20%



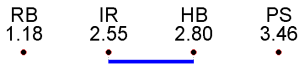
(e) 30%



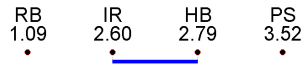
(f) 40%



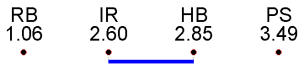
(g) 50%



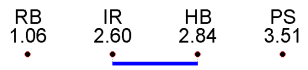
(h) 60%



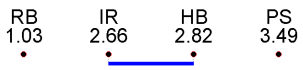
(i) 70%



(j) 80%



(k) 90%



(l) 100%

Figure 2.8: Nemenyi post-hoc tests per test scenario at α=0.05 (e.g., NB)

2.5 Summary

This chapter proposed a structured approach for obtaining calibrated probability under changes in positive class prevalence. Obtaining the calibrated probability is an effective way of dealing with class imbalance and class overlap problems because most conventional algorithms focus on classification of the major sample while ignoring or misclassifying the minority sample. Further, direct application of a classification model's outcome may result in substantial error. Consequently, this chapter proposed the robust RB calibration technique, which obtains accurate calibrated probability even when the test and training sets have different prevalence of the positives. The proposed method uses TPR and FPR, which are insensitive to class skews, and directly reflect the prevalence of the positives in the test set. In addition, the method differentiates the distribution nature within a class value from the effect of the class prevalence by using ROC curves to give performances that are robust to class skews.

To verify the effectiveness of RB, we conducted experiments with five classification algorithms and 87 binary-class datasets under conditions of changing positive class prevalence in the test sets. We evaluated RB and the well-known calibration methods HB, PS, and IR in terms of BS and CL. Among the evaluated calibration methods, the performance of the proposed RB was the most outstanding across all the binary-class datasets. The superiorly of the RB is decisive in terms of average, number of best performed datasets, and average rank. In addition, the performance of RB across 12 different test set prevalence conditions was the best. In cases where the positive prevalence of the test set

was identical to that of training set, however, the performance of RB was lower than that of IR, but without significant difference. RB also exhibited the best performance under 11 other test conditions, often exhibiting statistically significant superiority. RB estimated the calibrated probability comparatively well regardless of any changes in the prevalence of the positive class—which is our key objective.

Considering these promising experimental results, we plan to apply the proposed calibration technique to real-world problems in future work. The proposed method can be applied to personnel management in fields requiring careful recruiting and monitoring such as the military. For example, estimating maladjustment probability is very important in predicting the management of persons subject to conscription. Different management practices, such as exemption from the military and designation as Custody Sergeant, could then be applied to subjects that have a higher maladjustment probability. Another example is estimation of the fault rate to facilitate quality management prediction of product lines. We can estimate the relative and absolute probabilities of faults for the products using the calibration method. Product groups with higher fault rates could then be recalled and inspected more thoroughly. The method can also be applied to medical diagnosis, text categorization, and risk/disease prediction, which all require effective calibration techniques, because it supports decision making with good assessment of cost and effect.

In RB, it is assumed that the prevalence of the positive class is known. Thus, reliable prediction of prevalence is necessary in order to obtain accurate calibrated probabilities. Recent studies have shown that the prevalence of the

positive class can be estimated relatively well even without a perfect classifier. Some prevalence estimation methods that perform well in previous studies have also been presented. However, more accurate prevalence estimation is required to boost the efficacy of RB.

Chapter 3

Application of ROC-based Binning for predicting maladjusted soldiers

3.1 Background

The military includes maladjusted conscripts such as the mentally ill, the suicidal, the imprisoned, and those determined by the military commander to be maladjusted. Because conscripts possessing certain personality traits are likely to have trouble in discharging their military duties, Korea's Military Manpower Administration uses the Military Personality Inventory (MPI), a type of personality test, to assess whether a person subject to military conscription is mentally fit for active service in the Republic of Korea (ROK) Armed Forces. The MPI was developed by the Ministry of National Defense and the Korean Psychological Association in 1997 and has been used as a personality test of those subject to physical draft since 1999. The MPI was originally used for screening eligible conscripts, but it has also been applied to identify conscripts likely to become maladjusted after enlisting.

We use the MPI to predict which of the conscripts are likely to become maladjusted. However, such a prediction using the MPI presents a kind of class

imbalance problem, where the majority of the recruits continue active service and the minority, who are maladjusted to military life, are discharged early from active service. Class imbalance is a situation where one class is represented by a large number of instances while the other is represented by only a few. Thus, the total maladjusted proportion is very low in the MPI data. It is also a kind of class overlap problem in that some subjects may share similar MPIs but have different class values. Because most algorithms focus on the classification of the major sample while ignoring or misclassifying the minority sample, class imbalance and class overlap decrease the performance of standard classification algorithms(Denil & Trappenberg, 2010; Japkowicz & Stephen, 2002). Therefore, direct application of classification algorithms might result in substantial error. Instead, the analytical framework of Receiver Operating Characteristics (ROC) can serve as a more suitable tool, because it helps estimate the maladjusted proportion among subjects sharing similar MPIs. Estimating the maladjusted proportion is an effective way to identify the real-world probability of a subject being maladjusted.

This study, therefore, proposes the effective utilization of the ROC curve using the MPI toward the management of persons subject to military conscription. In Section 3.2, we briefly describe the MPI used by the Korean military. In Section 3.3, we apply various classification algorithms to the MPI test results and review their accuracy. In Section 3.4, we propose our method for estimating the maladjusted proportion using the ROC curve and analyse its useful applications. In Section 3.5, we conclude and discuss the contribution of this study and future work. Lots of this chapter's contents were published in Journal of

Applied Statistics (M. Sun et al., 2015).

3.2 Military Personality Inventory Data

This study uses a real-world MPI dataset of persons subjected to military conscription from January 2003 to May 2007 in the ROK. The MPI is a personality test used to assess mental health (Xiong et al., 2005). Since the ROK military adopted the conscription system, all men are examined for conscription by the Military Manpower Administration once they become 19 years of age. All examinees for conscription are tested using the MPI, the object of which is to determine the subject's personality and estimate the likelihood of his becoming maladjusted during active service. The obtained MPI dataset contains 4,456 subjects, each of whom responded to 151 'yes/no' type questions. The results of the MPI dataset indicate that there were 577 maladjusted and 3,879 well-adjusted conscripts (12.9% and 87.1% of the total), respectively. The data were collected and maintained by Korea's Military Manpower Administration and the Korea Institute for Defense Analysis (Choi et al., 2009). The institute has taken care of analysis and improvement for MPI test in the ROK Armed Forces since 2007. The test measures are based on expert reviews and results of t-test for the equality of means of maladjusted and well-adjusted conscripts from previously used or suggested measures by a psychologist and military personnel. Currently, 10 measures are used to screen eligible conscripts to the ROK Armed Forces. Linear Discriminant Analysis (LDA) is applied to the results of the MPI test in order to select subjects who are likely to be maladjusted. They should

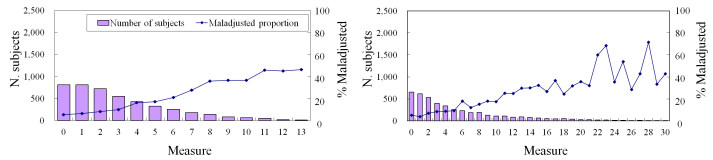
have further psychiatric testing to ascertain their eligibility for exemption from military service. The 10 measures of MPI are divided into 3 categories: neurosis, psychosis, and accident. Table 3.1 provides information about each measure. The terms 'neurosis' and 'psychosis' are both used to describe conditions or illnesses that affect mental health. The primary difference between the two is the manner in which they affect mental health (Goldberg, 1965). Neurotic behaviour can be naturally present in any person with a developed personality. Psychotic behaviour can be triggered intermittently by various influences. The last category, 'accident' measures the potential of the conscript to undertake improper actions or cause trouble during military service.

The MPI data show a class overlap in that persons sharing similar MPI test results include both the maladjusted and the well-adjusted.Figure 3.1 shows the distribution of the subjects and the maladjusted proportion by measure. The larger the value of each measure, the lower the mental health and the higher the possibility of being among the maladjusted. The majority of the subjects present the smaller value in each measure. Most values of a measure record high frequency for the well-adjusted. Figure 3.2 shows First and second PCA of the MPI dataset. F2 of all attritutes and first PCA are 0.958 and 0.858 respectively.

The MPI data are fairly imbalanced, and it is notable that the proportion of maladjusted conscripts among the entire body of conscripts is less than 12.9% in the real world. Class imbalance problems, such as those indicated by the MPI, are likely to induce significant deterioration in the accuracies achieved by existing learning and classification systems (P.-N. Tan et al., 2006).

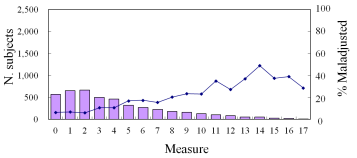
Table 3.1: Measures and numbers of relevant questions in the MPI

Category	Measure	Characteristics	N. questions	
			yes	no
Neurosis	Anxiety	Feelings of tension and worried thoughts	10	3
	Depression	Lack of pleasure in daily activities	28	4
	Somatization	Physical complaints for which no physical causes can be found	15	2
	Personality disorder	Permanent disposition to behaviour in ways causing suffering to oneself or others	16	9
Psychosis	Schizophrenia	Incoherent or illogical thoughts, bizarre behaviour and speech, and delusions or hallucinations	13	0
	Paranoia	Systematized delusions and the projection of personal conflicts, which are ascribed to the hostility of others	10	4
Accident	Desertion	Action of leaving military service or duty without the intention of returning	10	3
	Adjustment disorder	Short-term condition that occurs when a person is unable to cope with, or adjust to, military service	13	10
	Behaviour retardation	Not achieving milestones within the time range of that normal variability	19	0
	Acting out	Expression of intrapsychic conflict or painful emotion through overt behaviour	24	0
	Total	193 questions (151 questions with de-duplication)		

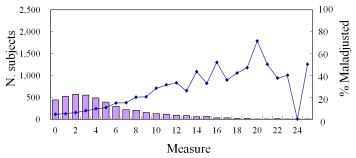


(a) Anxiety

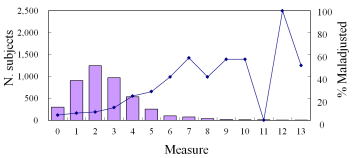
(b) Depression



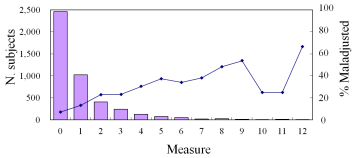
(c) Somatization



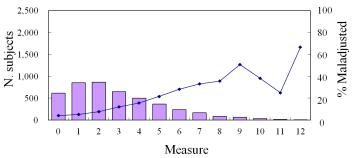
(d) Personality disorder



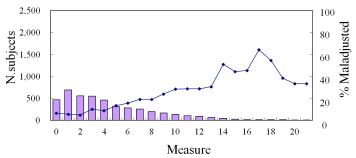
(e) Schizophrenia



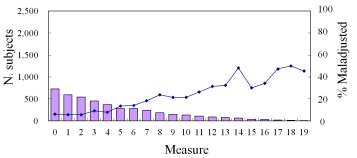
(f) Paranoia



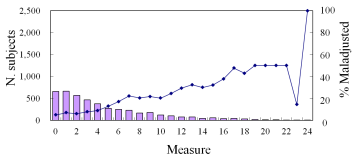
(g) Desertion



(h) Adjustment disorder



(i) Behaviour retardation



(j) Acting out

Figure 3.1: Distributions of the subjects (histograms) and maladjusted proportions (lines) of TPR Binning

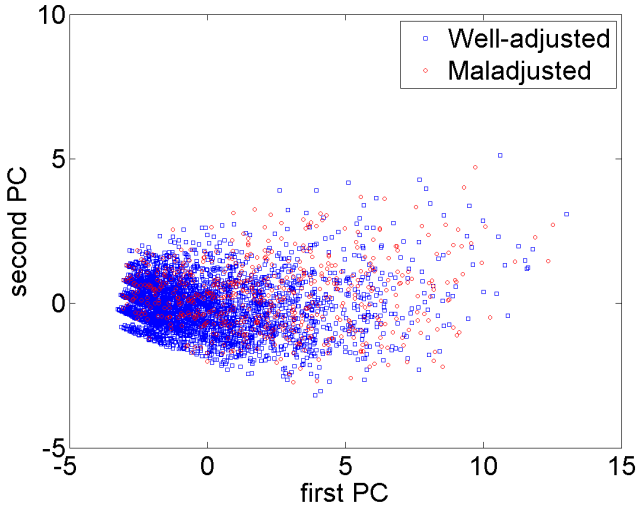


Figure 3.2: First and second PCs of the MPI dataset

3.3 Performance of classification algorithms using the MPI dataset

We apply various classification algorithms to the MPI dataset to check how accurately they classify subjects as maladjusted or well-adjusted. We use well-known classification algorithms such as NB, QDA, LDA, LR, and SVM. Given subject $\mathbf{x}$ of d-dimensional vector $(x_1 \ x_2 \ \ldots x_d)^T$, Bayesian classification algorithms allows us to express the posterior probability in terms of the evidence, class conditional probability, and prior probability for each class value c:

$$p(c/\mathbf{x}) = \frac{p(\mathbf{x}/c)p(c)}{p(\mathbf{x})}. \tag{3.1}$$

Among the NB, QDA, and LDA algorithms, the class conditional probability assumes a Gaussian distribution with different covariance. LDA assumes a

common class-independent covariance, while QDA assumes a class-dependent covariance. The NB algorithm is equivalent to QDA with a diagonal covariance matrix. LR estimates $f(\mathbf{x})$, that is, the probability that subject $\mathbf{x}$ belongs to a positive class, using the logit function:

$$f(\mathbf{x}) = \frac{e^z}{e^z + 1} \text{ where } z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_d x_d. \tag{3.2}$$

Given a training set of N data points, where $\mathbf{x}_i$ is the i-th subject pattern, and y_i is the i-th class pattern, the support vector method constructs a maximum margin classifier of the form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}^T \mathbf{x} + b \text{ where } y_i \in \{-1,\ 1\}. \tag{3.3}$$

SVM models have a cost parameter, C, that controls the trade-off between allowing training errors and forcing rigid margins. Here, C is selected using a grid search from candidates [0.5 1.0 ... 9.5 10.0].

There are four possible metrics: if a subject is maladjusted and is classified as such, he is counted as a True Positive (TP); if classified as well-adjusted, he is counted as a False Negative (FN). If the subject is well-adjusted and is classified as such, he is counted as a True Negative (TN); if classified as maladjusted, he is counted as a False Positive (FP).

We check the metrics of various classification algorithms for the MPI dataset (Table 3.2). All data processing and algorithm implementation were performed off-line using the commercial software package MATLAB (*Matlab version 7.10.0*, 2010). SVM was implemented from LIBSVM (Chang & Lin, 2011). We define the maladjusted as the positive class and the well-adjusted as the negative class

and estimate the metrics using five-fold cross validation (P.-N. Tan et al., 2006). The five-fold cross validation method segments the data into 5 equal-sized partitions, and each partition is used exactly 4 times for training and once for testing. During each run, one of the partitions is chosen for testing, and the rest, for training. This procedure is repeated 5 times, and total performance is assessed by summing up the results for all 5 runs. As the accuracy of every classification algorithm ranges from 0.808 to 0.871, they perform fairly well. However, because of class imbalance, the TPR of every classification algorithm is very low (0.000 to 0.484).

Table 3.2: Performance of classification algorithms applied to the MPI dataset

Classifier	Acc.	$\hat{P}$ (TP+FP)	$\hat{N}$ (FN+TN)	TPR	FPR	FNR	TNR
NB	0.808	836	3,620	0.484	0.144	0.516	0.856
QDA	0.836	619	3,837	0.404	0.100	0.596	0.900
LDA	0.867	196	4,260	0.152	0.028	0.846	0.972
LR	0.868	134	4,322	0.108	0.019	0.892	0.981
SVM	0.871	0	4,456	0.000	0.000	1.000	1.000

3.4 Use of ROC curve to obtain calibrated probability

We estimate the proportion of the maladjusted in subject groups with similar MPI test results using the ROC curve, which is widely used in machine learning and the medical field.

3.4.1 Generation of ROC curve

Various classification algorithms are used to validate the proposed method and to check which algorithm is the most suitable. Some classification algorithms, such as NB, QDA, LDA, LR, and SVM, naturally yield probabilistic values when applied on a data and represent the degree to which a subject is a member of a class. We can calculate the *score*, the numeric value used directly for class prediction, by combining the function values. In the case of NB, QDA, and LDA that provide two kinds of function values, p(maladjusted/$\mathbf{x}$) and p(well-adjusted/$\mathbf{x}$), the score, p($\mathbf{x}$=maladjusted), is calculated as p(maladjusted/$\mathbf{x}$) divided by the sum of p(maladjusted/$\mathbf{x}$) and p(well-adjusted/$\mathbf{x}$), and ranges from 0 to 1. Whereas LR and SVM algorithms yield a single function that computes the degree to which a subject belongs to a class, and here, the functional value is directly used as the score.

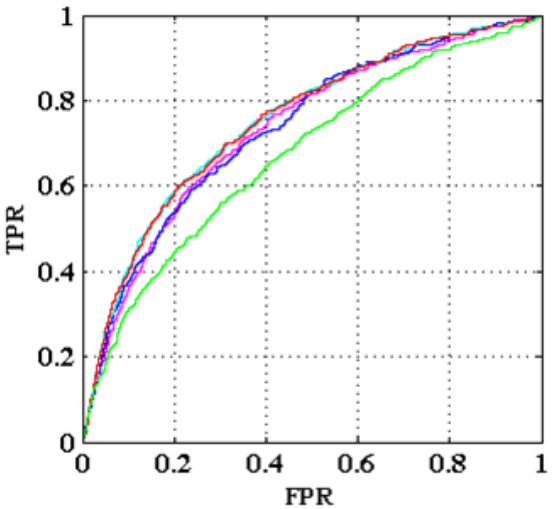


Figure 3.3: ROC curves of classification algorithms applied to the MPI dataset.

Figure 3.3 shows the ROC curves of various classification algorithms for the MPI dataset. These curves are quite similar for the abovementioned classification algorithms except the SVM algorithm, unlike their classification accuracies, which differ greatly between all classification algorithms.

Table 3.3: Comparison of AUC values of classification algorithms on the MPI dataset.

Classifier	AUC	p-value	95% CI	p-value(pair-wise comparison)			
				QDA	LDA	LR	SVM
NB	0.735	< 0.001	$0.722 \sim 0.748$	0.895	0.002	0.004	< 0.001
QDA	0.735	< 0.001	$0.722 \sim 0.748$		0.001	0.003	< 0.001
LDA	0.753	< 0.001	$0.740 \sim 0.766$			0.281	< 0.001
LR	0.752	< 0.001	$0.739 \sim 0.764$				< 0.001
SVM	0.675	< 0.001	$0.661 \sim 0.689$				

The Area Under the ROC Curve (AUC) summarizes the performance of the classification algorithm (Martínez-Camblor et al., 2011). Following Delong's proposal (DeLong et al., 1988), we test the significance of individual AUC values and the paired comparison. Table 3.3 shows that the AUC values range from 0.675 to 0.753, and all the classification algorithms have the ability to distinguish between the two classes with a very small p-value (< 0.001). The LDA has the largest AUC value, which is significantly different from those of the other classification algorithms, except LR, where the significance level is greater than 0.05. The SVM algorithm shows the smallest AUC, which is significantly lower than the AUCs of the other classification algorithms. The statistical test was

performed off-line using the software package MedCalc for Windows (*MedCalc version 14.8.1*, 2014).

3.4.2 Obtaining calibrated probability

Point (0, 1) on the ROC curve represents perfect classification, and no classification algorithm produces perfect performance. However, subjects having large scores are more likely to be maladjusted, and those having similar scores also share a similar probability to be maladjusted. We suggest a method to calculate the maladjusted proportion for each group having similar scores. To determine the score interval for each group, we use the TPR information obtained from the ROC curve. On the ROC curve, a single subject is matched with a single ROC point (FPR, TPR) according to his score and then allocated to a group on the basis of the matched TPR. Consequentially, we divide the subjects into several groups depending on their scores and estimate the maladjusted proportion in each group.

3.4.2.1 Application of ROC Binning and the other calibration methods

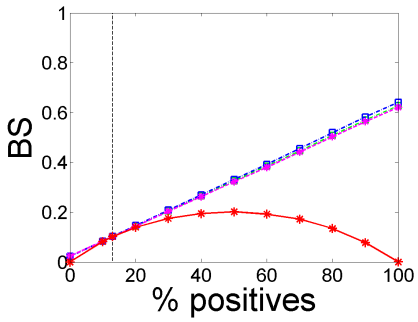
For the MPI dataset, BS was evaluated for 12 test conditions using 5 classification algorithms as base-learner. ROC Binning represented best performance across different test prevalence. Figure 3.4 shows BS distribution by test condition for each classification algorithm. We can check that all classification algorithms resulted in similar patterns. ROC Binning represented best performance across different maladjusted prevalence for every classification algorithm. The

BS of the previously proposed calibration methods greatly increased as prevalence increased. On the other hand, in ROC Binning, BS remained stable or even decreased.

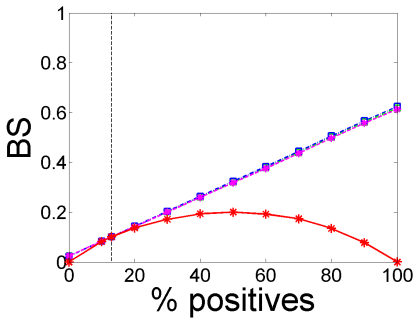
In this study, the BS was averaged from 12 repetitions (12 test conditions) for each classification algorithm. The averaged BS of each calibration method is shown in Table 3.4. We can check that all classification algorithms resulted in similar characteristics. ROC Binning showed best performance and Isotonic Regression showed second best. Difference between them ranged from 0.170 to 0.199 across the entire classification algorithms.Table 3.5 shows the summary of BS per test condition over the entire datasets when using LDA.

Table 3.4: BS for all the test conditons on the MPI dataset

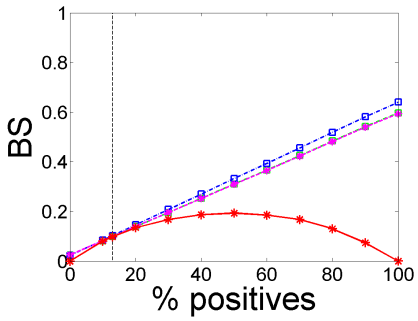
Classfier	HB	PS	IR	RB
NB	0.307	0.313	0.304	0.123
QDA	0.303	0.306	0.301	0.122
LDA	0.293	0.313	0.292	0.118
LR	0.29	0.309	0.29	0.12
SVM	0.331	0.364	0.333	0.134



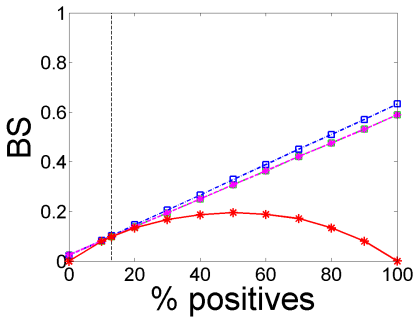
(a) NB



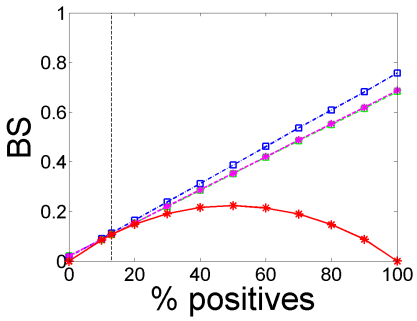
(b) QDA



(c) LDA



(d) LR



(e) SVM

Figure 3.4: BS on the MPI dataset

Table 3.5: BS per test conditon on the MPI dataset (e.g., LDA)

% positives	HB	PS	IR	RB
p(training)	0.099	0.103	0.099	0.099
0	0.025	0.023	0.025	0.000
10	0.082	0.085	0.082	0.080
20	0.139	0.147	0.139	0.134
30	0.196	0.208	0.196	0.168
40	0.253	0.27	0.252	0.187
50	0.311	0.333	0.310	0.193
60	0.367	0.394	0.365	0.186
70	0.425	0.456	0.423	0.168
80	0.484	0.519	0.482	0.130
90	0.542	0.581	0.540	0.074
100	0.597	0.64	0.594	0.000

3.4.2.2 TPR Binning and its application

Using the MPI dataset, we additionally calculate the maladjusted proportion in each group formed using the TPRs seen on the ROC curve. We name it as TPR Binnig. We compare the maladjusted proportion of each group in the training datasets with that in the test datasets using five-fold cross validation to predict the performance of the suggested method. The cross validation proceeds in two phases: the training phase and the test phase. During each run of the five-fold cross validation, 80% of the MPI dataset is used for training, and the remaining 20% is used for the test.

In the training phase, we divide the training subjects into 10 groups and

calculate the maladjusted proportion in each group (Figure 3.5). We construct a classifier and generate the ROC curve using the training datasets. We divide training subjects into 10 groups formed based on the TPRs. The TPRs are divided into 10 groups with Equal Width Interval Discretization (Dash et al., 2011). We set the interval to 0.1, because the TPRs range from 0 to 1. Groups 1 to 9 have half-closed intervals containing only minimum points, and the last group (Group 10) has closed intervals containing both its minimum and maximum points. We calculate the maladjusted proportion in each group k (MP_k) based on the score range ((3.4)), namely $(b\_score\ (l)_k,\ b\_score\ (u)_k]$ or the lower bound and upper bound of a bin respectively, of scores corresponding to the TPRs of group k.

$$MP_k = \frac{\text{maladjusted subjects in } (b\_score\ (l)_k,\ b\_score\ (u)_k]}{\text{subjects in } (b\_score\ (l)_k,\ b\_score\ (u)_k]} \tag{3.4}$$

In the test phase, we divide the test subjects into 10 groups by adopting the classifier constructed in the training phase. Then, we calculate the maladjusted proportion in each subject group. We generate the scores of the test subjects using the classifier constructed in the training phase. Thus, we divide the test subjects into 10 groups based on the score ranges for the groups obtained in the training phase. Each group is expected to have the same TPR as the corresponding group of training subjects. For example, if the training subjects in the group corresponding to TPRs of [0.1, 0.2) have scores of (0.8, 0.9], the test subjects having the scores of (0.8, 0.9] will also belong to the group with the expected TPRs of [0.1, 0.2). For TPR Binning, BS was evaluated for 12 test

conditions. Table 3.6 shows BS distribution by test condition when using each classification algorithm. TPR Binning represented similar performances with ROC Binning across different test prevalence. The averaged BS from 12 repetitions (12 test conditions) is shown in Table 3.6. We can check that there is little difference in performance between TPR Binning and ROC Binning for every classification algorithm. In summary, TPR Binning and ROC Binning showed equivalent results for MPI dataset where the class imbalance and class overlap is severe and thus classification performance is low. Table 3.7 shows the summary of BS per test condition over the entire datasets when using LDA.

Table 3.6: BS of TPR Binning for all the test conditions on MPI dataset

Classfier	HB	PS	IR	RB	TB
NB	0.307	0.313	0.304	0.123	0.122
QDA	0.303	0.306	0.301	0.122	0.123
LDA	0.293	0.313	0.292	0.118	0.119
LR	0.290	0.309	0.290	0.120	0.120
SVM	0.331	0.364	0.333	0.134	0.134

This study conduct further analysis for TPR Binning in the situation where % total maladjusted(training)=% total maladjusted (test).We quantify the MP_i differences between the training datasets and test datasets by calculating the Mean Square Error (MSE) and Mean Absolute Error (MAE), both of which consider the percentage of subjects belonging to each group in the test datasets, with the percentage of the $subjects(test)_i$ serving as the weight value. The MSE and MAE are calculated using Eq. (3.5) and Eq. (3.6) respectively, where

Inputs: $\{(\mathbf{x}_i, y_i), i = 1, ..., N\}$, the training set; $\{\mathbf{x}_j, j = 1, ..., N'\}$, the test patterns; $p(positive)$, the positive class prevalence of test set; M, number of bins

Output: $(\{\hat{p}_{TB}(\mathbf{x}_{b_k}), k = 1, ..., M\}$, the calibrated probability for patterns in each bin using TPR Binning

// Training phase:

Step 1: Construct a classifier from the training dataset

$Classifier \leftarrow$ a leaning model that fits the relationship between $\mathbf{x}$ and y

Step 2: Calculate the training pattern's score by employing the Classifier

For each pattern $i = 1 : N$

$score_i \leftarrow (f\_Scoring \otimes Classifier)(\mathbf{x}_i)$

Step 3: Generate ROC points from a finite training set

For each pattern $i = 1 : N$

$(tpr_i, fpr_i, threshold_i) \leftarrow$ ROC point when $score_i$ is the threshold

Step 4: Divide TPR into 10 bins with Equal Width Interval

$\{[0, tpr_1), ..., [tpr_9, tpr_{10}]\} \quad \leftarrow \{[0, 0.1), ...[0.9, 1]\}$

Step 5: Find the parameters (b_TPR, b_FPR, b_SCORE) for ROC Binning

For each bin $k = 1 : 10$

$(b\_tpr(l)_k, b\_tpr(u)_k) \leftarrow \{tpr_i : \ tpr_i \in [tpr_{k-1}, tpr_k)\}$

$(b\_fpr(l)_k, b\_fpr(u)_k) \leftarrow \{fpr_i : \ tpr_i \in [tpr_{k-1}, tpr_k)\}$

$(b\_score(l)_k, b\_score(u)_k) \leftarrow \{\text{threshold}_i : \ tpr_i \in [tpr_{k-1}, tpr_k)\}$

// Test phase:

Step 6: Calculate the test pattern's score by employing the Classifier

For each pattern $j = 1 : N'$

$score_j \leftarrow (f\_Scoring \otimes Classifier)(\mathbf{x}_j)$

Step 7: Estimate the calibrated probability for patterns belonging to a bin, k

For each bin $k = 1 : M$

$\mathbf{x}_{b_k} \leftarrow \{\mathrm{x}_j : \ score_j \in (b\_score(l)_k, b\_score(u)_k)\}$

$\hat{p}_{TB}(\mathbf{x}_{b_k}) = \frac{(b\_tpr(u)_k - b\_tpr(l)_k) \times p(positive)}{(b\_tpr(u)_k - b\_tpr(l)_k) \times p(positive) + (b\_fpr(h)_k - b\_fpr(l)_k) \times (1 - p(positive))}$

Figure 3.5: Computation algorithm utilized by TPR Binning

Table 3.7: BS of TPR Binning per test condition on MPI dataset (e.g., LDA)

% positives	HB	PS	IR	RB	TB
p(training)	0.099	0.103	0.099	0.099	0.099
0	0.025	0.023	0.025	0.000	0.000
10	0.082	0.085	0.082	0.080	0.081
20	0.139	0.147	0.139	0.134	0.135
30	0.196	0.208	0.196	0.168	0.168
40	0.253	0.270	0.252	0.187	0.188
50	0.311	0.333	0.310	0.193	0.195
60	0.367	0.394	0.365	0.186	0.188
70	0.425	0.456	0.423	0.168	0.171
80	0.484	0.519	0.482	0.130	0.132
90	0.542	0.581	0.540	0.074	0.076
100	0.597	0.640	0.594	0.000	0.000

$MP(training)_k$ and $MP(test)_k$ stand for the MP_k in the training datasets and the test datasets respectively.

$$MSE = \sum_{k=k}^{10} (MP(training)_k - MP(test)_k)^2 \times \text{percentage of } subjects(test)_k \quad (3.5)$$

$$MAE = \sum_{k=1}^{10} |MP(training)_k - MP(test)_k| \times \text{percentage of } subjects(test)_k \quad (3.6)$$

The MSEs and MAEs range from 0.0009 to 0.0022 and 0.021 to 0.029 respectively for all the classification algorithms, and the NB algorithm has the smallest MSE and MAE (Table 3.8). Figure 3.6 shows the number of subjects in each group of the test datasets and the maladjusted proportions in each group of the training and test datasets. The maladjusted proportions in each group of the

training and test datasets are a little different. All the classification algorithms represent a linear relation (negative correlation) between the TPRs and maladjusted proportions; the latter decreases as the former increases. But a few TPRs, such as [0.1, 0.2) of the QDA, depart from the trend. The number of subjects increases as the TPRs increase.

Table 3.8: MSEs and MAEs of the maladjusted proportion of TPR Binning on the MPI dataset

Classfier	MSE	MAE
NB	0.0009	0.021
QDA	0.0022	0.028
LDA	0.0017	0.025
LR	0.0018	0.027
SVM	0.0019	0.029

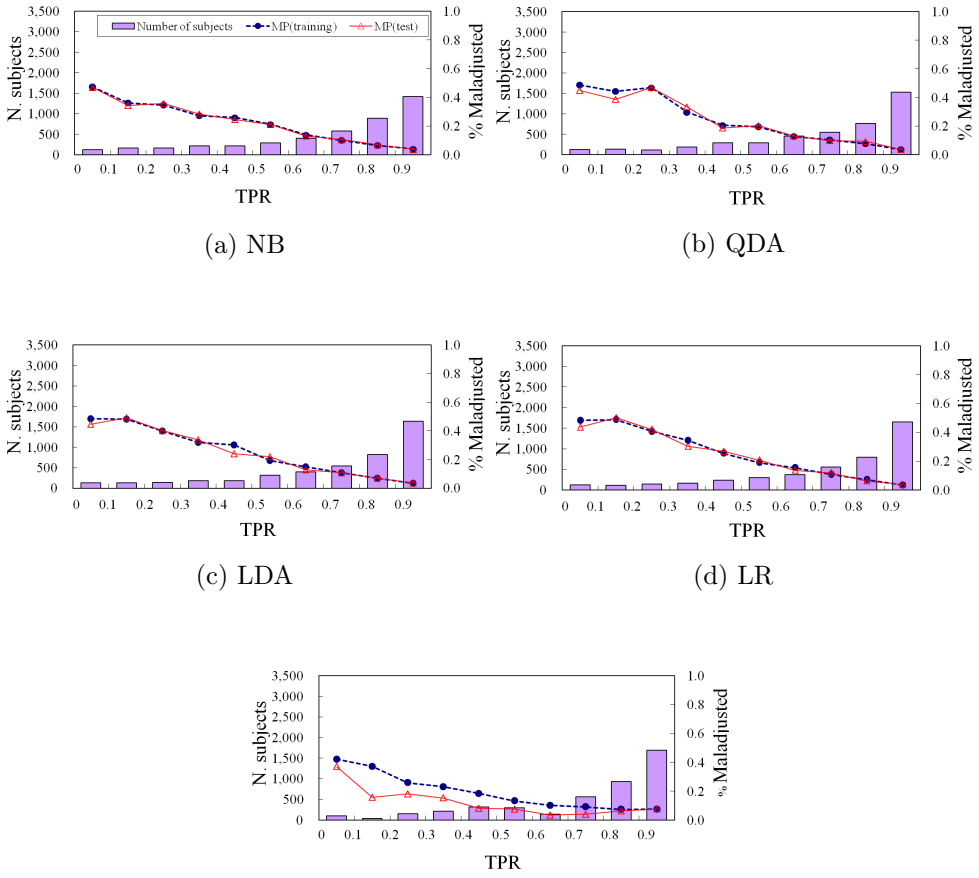
We demonstrate the effectiveness of the proposed method using the LDA and NB algorithms, which provided the best performance for the AUC and error respectively. In managing persons subject to conscription, we can determine a reference value of TPRs to efficiently screen out persons requiring either exemption from active military service or special attention during active service. To determine a reference value, we can conduct a Pareto analysis by calculating the cumulative percentage of maladjusted subjects and total subjects. Pareto analysis is a statistical technique in decision making used to select a limited number of conditions producing a significant overall effect (Amoroso, 1938). Table 3.9 shows the cumulative percentage of maladjusted subjects and all

subjects in both the training and test datasets after sorting the groups by ascending order of TPRs using the LDA and NB algorithms. As the TPRs of a group increase, the percentages of the maladjusted remain constant, and the percentages of the subjects increase significantly. Subjects whose TPRs are less than 0.3 comprise about 30% of all maladjusted subjects and about 10% of all subjects when using the NB algorithm; if we randomly select 10% of all subjects, they would include 10% of all maladjusted subjects. Thus, when using the Pareto analysis with the ROC curve, we can efficiently choose those persons who are probably maladjusted.

Table 3.9: Cumulative % of the maladjusted subjects and subjects in the training and test sets on the MPI dataset

TPRs	LDA				NB			
	Maladjusted subjects		Subjects		Maladjusted subjects		Subjects	
	Training	Test	Training	Test	Training	Test	Training	Test
[0, 0.1)	10.0	9.7	2.7	2.8	10.0	10.1	2.8	2.8
[0.1, 0.2)	20.0	20.3	5.4	5.6	20.0	19.8	6.4	6.4
[0.2, 0.3)	30.0	30.0	8.6	8.8	30.0	29.6	10.1	10.1
[0.3, 0.4)	40.0	40.4	12.7	12.7	40.0	40.4	14.8	14.9
[0.4, 0.5)	50.0	48.0	17.0	16.8	50.0	49.7	19.8	19.8
[0.5, 0.6)	60.0	60.0	23.8	23.8	60.0	60.3	26.0	26.3
[0.6, 0.7)	70.0	69.2	32.5	32.7	70.0	69.3	35.5	35.3
[0.7, 0.8)	80.0	79.2	44.6	44.8	80.0	79.9	48.2	48.2
[0.8, 0.9)	90.0	89.3	63.5	63.2	90.0	90.6	68.0	68.2
[0.9, 1.0]	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

The MPI test currently selects about 7% of the subjects for conscription for further psychiatric diagnosis to determine their eligibility for exemption



(a) NB

(b) QDA

(c) LDA

(d) LR

(e) SVM

Figure 3.6: Distributions of the subjects (histograms) and maladjusted proportions (lines) of TPR Binning on the MPI

from military duty. Our calculations indicate that a maximum of 25.3% of the maladjusted can be exempted from active military service through psychiatric diagnoses of the top 7% of the subject scores using the LDA. For the NB algorithm, a maximum 21.7% of the maladjusted can be exempted from active

military service. The results of the other classification algorithms, the QDA, LR, and SVM, show that the subjects having the top 7% scores comprise 23.4%, 24.6%, and 18.3% respectively of all maladjusted subjects.

NB and LDA inherently help us estimate the probability of a subject being maladjusted, but the probability needs to be calibrated to represent the actual occurrence of the maladjusted subject. The probability of a subject's being maladjusted, $p(\mathbf{x}=$ maladjusted), is calculated using

$$p(\mathbf{x} = \text{maladjusted}) = \frac{p(\text{maladjusted}/\mathbf{x})}{p(\text{maladjusted}/\mathbf{x}) + p(\text{well-adjusted}/\mathbf{x})}. \tag{3.7}$$

We compare the maladjusted proportion with the maladjusted probability in each group for the LDA and NB algorithms (Table 3.10). If the maladjusted probability calculated by the classification algorithm represents the actual occurrence of the maladjusted subjects, the maladjusted probability and the maladjusted proportion would be almost identical. However, there are large gaps in practical terms between the two metrics. Therefore, we cannot directly employ the maladjusted probability estimated by a classification algorithm as the actual proportion of the maladjusted; we need to compute the actual maladjusted proportion using the ROC curve.

The percentage of $subjects(test)_i$ could be estimated using the *estimated* $MP(test)_i$ and $TMP(test)$ (Eq. (3.8)), as the percentage of the maladjusted subjects belonging to each group in the test datasets can be assumed to be constant at 10% (as in the training datasets).

$$\text{Estimated percentage of } subjects(test)_k = \frac{TMP(test) \times 0.1}{\text{Estimated } MP(test)_k} \tag{3.8}$$

Table 3.10: Comparison of the maladjusted proportion with the maladjusted probability p(**x**=maladjusted) on the MPI dataset

TPRs	LDA			NB		
	p(**x**=maladjusted)	Calibrated probability		p(**x**=maladjusted)	Calibrated probability	
		Training	Test		Training	Test
[0, 0.1)	0.751	0.487	0.446	1.000	0.474	0.470
[0.1, 0.2)	0.524	0.483	0.491	1.000	0.361	0.343
[0.2, 0.3)	0.379	0.400	0.403	0.999	0.347	0.359
[0.3, 0.4)	0.257	0.319	0.331	0.972	0.273	0.284
[0.4, 0.5)	0.199	0.301	0.250	0.654	0.259	0.245
[0.5, 0.6)	0.148	0.194	0.219	0.165	0.212	0.211
[0.6, 0.7)	0.104	0.149	0.139	0.020	0.137	0.130
[0.7, 0.8)	0.071	0.108	0.109	0.003	0.102	0.105
[0.8, 0.9)	0.044	0.068	0.069	0.000	0.066	0.070
[0.9, 1.0]	0.008	0.035	0.038	0.000	0.040	0.037

3.5 Summary

This study proposed a novel application of the ROC curve to cope with class imbalance and class overlap problems. In class imbalance problems, direct application of classification algorithms might result in substantial error, because most algorithms focus on the classification of the major sample while ignoring or misclassifying the minority sample. The accuracy for 5 of the classification algorithms to which we applied the class-imbalanced MPI dataset ranged from 0.808 to 0.871, and the TPRs were severely low (0.000 to 0.484). Among them, the SVM algorithm showed a TPR of 0, because it is vulnerable to class imbalance. The ROC curve can serve as a more suitable tool, because it helps estimate the actual probability of maladjusted conscripts. The ROC curve is a

well-established statistical concept that shows insensitiveness to changes in the total positive class proportion. Using a real-world MPI dataset, we estimated the maladjusted proportion of persons sharing similar MPI test results. The elaborate methodology divided subjects into 10 groups based on the classifier's scores and estimated the maladjusted proportion exactly in each group. To set the score interval for each group, we used information about the TPR obtained from the ROC curve.

This study applied the proposed ROC Binning and TPR Binning into MPI datasets BS was evaluated for 12 test conditions using 5 classification algorithms as base-learner. We can check that all classification algorithms resulted in similar characteristics. ROC Binning represented best performance across different test prevalence. There is little difference in performance between TPR Binning and ROC Binning for every classification algorithm. In summary, TPR Binning and ROC Binning showed equivalent results for MPI dataset where the class imbalance and class overlap is severe and thus classification performance is low.

Estimating the maladjusted proportion is very useful in predicting the management of persons subject to conscription. Different management practices would be applied to each subject group by checking the corresponding maladjusted proportion. Subject groups with greater maladjusted proportions require further psychiatric testing to ascertain their eligibility for exemption from military service. If subjects with greater maladjusted proportions enter military service, they should be provided with additional attention during service.

Various classification algorithms were used to validate the proposed method

and to check which algorithm is the most suitable. For every classification algorithm, the AUC was significantly different from 0.5, and the WMSEs and WMAEs were very small. The LDA and NB algorithms provided the best performance in terms of the AUC and error respectively. The SVM algorithm showed the smallest AUC, which was significantly lower than the AUCs of the other classification algorithms. Besides, the results for all the tested classification algorithms showed that the subjects having the top 7% scores comprised 18.3% to 25.3% of all maladjusted subjects, and LDA and SVM were the best and worst performing algorithm respectively. On the whole, LDA was the most appropriate algorithm for the proposed method with the MPI dataset. On the other hand, the SVM algorithm was inappropriate for the proposed method with the MPI dataset.

Chapter 4

Similarity-based Adjusted Count (SAC) method

4.1 Background

The analytical framework of prevalence estimation is a useful tool for the identification of the total positive proportion of target data because it provides the relevant aggregate information. It is a particularly significant issue in class imbalance (Duman et al., 2012) and class overlap (Denil & Trappenberg, 2010) problems, where a direct application of the predictions of the classification model might yield substantial error. The aim of this task is to accurately estimate the class prevalence of a test set, assuming a training set with noticeably different prevalence distribution from that in a test set (Barranquero et al., 2013). A perfect classifier is sufficient, but not necessary, to satisfactorily estimate prevalence (Forman, 2008). From an intuitive perspective, prevalence estimation can provide more credible information than classification because it does not involve delivering accurate predictions on each instance. This is because the uncertainty in this case shifts from individual cases to their aggregate, and thus decreases. Even in case of an imperfect classifier, prevalence can be

satisfactorily estimated by using the performance measure of the classifier. For example, if the number of false positives balances that of false negatives, the number of predicted positives is equal to their actual number of occurrences in the final outcome.

Predicting prevalence for a dataset is useful to track trends over time and identify the differences among targets. For example, a manufacturing company can estimate the annual number of items that will be faulty in a production line without having to predict and identify the particular items that will be defective. The items manufactured on production lines with higher fault rates should be inspected more thoroughly to find ways to reduce faults. Moreover, medical diagnoses, text categorization, and risk/disease prediction require prevalence estimation because it supports decision making based on relative and absolute priorities in terms of cost effectiveness.

Research on prevalence estimation has hitherto been scant. Furthermore, differences in class prevalence between the training and test data can have a significant impact on the performance of conventional machine learning algorithms. However, data mining tasks usually assume that the training and test samples are obtained from the same population, and do not take into account prevalence difference (P.-N. Tan et al., 2006). A few studies have shown that classifiers can be used to estimate class prevalence and monitor changes or trends in distributions(Barranquero et al., 2013; Forman, 2008). This task, called quantification, deals with the prediction of the prevalence of the class of interest (Baccianella et al., 2013; Swets, 1988) with the objective of accurately estimating the prevalence of a positive class in the test set given a training set

with prevalence different from that of the test set.

The adjusted count (AC) method has been primarily used to estimate prevalence in past research because of its effectiveness. In AC-based quantifiers, the true positive rate (TPR) and the false positive rate (FPR) (Fawcett, 2006) are required to perform prevalence correction based on classification performance. The TPR and FPR were determined by many-fold cross-validation of the training dataset in past research. This implies that there should be no fundamental difference in the TPR and the FPR between the training and the test sets in order to ensure high quality of AC-based estimation. However, there is a difference in data distribution between the training and test sets in real-world scenarios, and this might cause changes in both the TPR and FPR to contaminate AC-based prevalence estimation with bias.

In this study, we propose a similarity-based adjusted count (SAC) to estimate prevalence by reflecting the difference in the characteristics of data distribution involving TPR and FPR between the training and the test sets. The SAC is an adaptive prevalence estimation method derived from the AC method, specifically by using the TPR and FPR of training instances similar to the test instances. This method fundamentally assumes that difference in data distribution between the training and test sets can affect classification performance, but similar instances in the feature space yield fairly similar performance scores. We use the k-nearest neighbors (k-NN) algorithm as base classification learner because it can remember the details of the topology of the data and search similar instances based on distance. In order to verify the effectiveness of the SAC, we compared it with other AC-base quantification methods using a support

vector machine or proportion-weighted k-NN as the base classification learner in terms of absolute error. Experimental results showed that our proposed SAC outperformed prevalent method on all binary class datasets.

The rest of this chapter is structured as follows. In Section 4.2, we briefly review past research on prevalence estimation by using classification algorithms as base learners. In Section 4.3, we detail our proposed SAC with weighted k-NN, which is adaptable to changes in data distribution in the test set. In Section 4.4, we describe experiments to test the performance of the SAC in comparison with other AC-based prevalence estimation methods on benchmark datasets. We offer our conclusions in Section 4.5.

4.2 Related work

Recall the setup for the binary classification problem, where we are given (a realization of) a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, ..., N\}$, where $\mathbf{x}_i$ represents an arbitrary attribute (often a vector in R^d) and $y_i \in \{-1, 1\}$ is the corresponding binary class label. The obvious method of estimating positive prevalence is to directly use the positive predictions of the classification model. Called Classify & Count (CC), it learns the classification model through the training set, and counts its positive class predictions on the test set to provide prevalence estimate p'_{CC} as

$$p'_{CC} = p' = \frac{\hat{P}}{N} \tag{4.1}$$

where N is the size of the test set and $\hat{P}$ is the cardinality of the set of positive predictions.

If the classification model is perfect, the CC produces accurate prevalence estimates. However, CC is not a satisfactory quantifier for imbalanced and overlapped data because a perfect classifier is unrealistic in these situations. With regard to bias in the CC method, Forman (Forman, 2008) has provided the following theorem and proof:

Theorem (Forman's theorem): For an imperfect classifier, the CC method will underestimate the true proportion of positives p in a test set for $p >$p*, and will overestimate it for $p <$ p*, where p* is the particular proportion at which the CC method estimates correctly, i.e., the CC method estimates p* exactly for a test set with p* positives.

Proof. The estimated prevalence p' of classifier predictions, written as a function of the actual positive prevalence p, is

$$p' = tpr \cdot p + fpr(1 - p). \tag{4.2}$$

If the classifier correctly estimates prevalence for a particular value p*, i.e., $p' = p^*$, then, for a strictly different prevalence $p^* + \Delta$, where $\Delta \neq 0$, the expected prevalence $p'(p^* + \Delta)$ is not equal to the true prevalence:

$$p'(p^* + \Delta) = tpr \cdot (p^* + \Delta) + fpr \cdot (1 - (p^* + \Delta) = p^* + (tpr - fpr) \cdot \Delta \tag{4.3}$$

Since an imperfect classifier is such that $(tpr - fpr) < 1$,

$$p'(p^* + \Delta) = \begin{cases} < p^* + \Delta \; if \; \Delta > 0, \\ > p^* + \Delta \; if \; \Delta < 0. \end{cases} \tag{4.4}$$

According to Forman's theorem, the CC method underestimates the actual prevalence when it is greater than p* and overestimates the actual preva-

lence when it is smaller than p^*. With the aim of correcting this bias, Forman (Forman, 2008) proposed a prevalence estimation method called adjusted count (AC) that employs a function from the actual positive prevalence to the expected positive prevalence through the classifier, as shown in Eq. (4.5):

$$p' == tpr \cdot p + fpr(1-p) \rightarrow p = \frac{p' - fpr}{tpr - fpr}. \tag{4.5}$$

In the AC method, TPR and FPR are obtained from a training set through a many-fold cross-validation to provide prevalence estimate p'_{AC} as

$$p'_{AC} = \frac{p'_{CC} - fpr_{train}}{tpr_{train} - fpr_{train}}. \tag{4.6}$$

By assumption, there should be no fundamental change in the TPR and the FPR between the training and testing sets for the AC method. This implies in turn that the TPR and the FPR should be independent of shifts in class prevalence to ensure satisfactory performance of the AC method. This assumption is satisfied in terms of statistical theory for specific conditions when the changes in class priors are obtained by means of stratified sampling (Fawcett & Flach, 2005; Webb & Ting, 2005).

The AC method can satisfactorily estimate class prevalence in many situations, but its performance degrades severely when the distribution of the training class is highly imbalanced (Forman et al., 2006). If a positive class is rarely encountered in the training set, the classifier will learn to almost always vote negatively for high accuracy, i.e., the value of the TPR is close to 0. The following small denominator in Eq. (4.6) of the expression for the AC method produces prevalence estimation vulnerable to variations in the estimation of the

TPR or the FPR. It renders the AC method highly sensitive to errors in the estimations of the TPR or the FPR.

Therefore, Forman also proposed alternative imbalance-tolerant methods based on the selection of thresholds for classification. The underlying intuition was that selecting a threshold that allows a sufficiently large denominator for the formula expressing the AC can yield a more robust prevalence estimate against estimation errors. Forman proposed four policies to select the threshold as shown Table 4.1. The first is the T50 method, which chooses the threshold where TPR=50%, hence avoiding the tail of the TPR curve. The second is the X method, which selects the threshold where the FPR is 1−TPR, thus avoiding the extreme values of both measures. The third method, the MAX method, chooses the threshold where TPR−FPR assumes the maximum value, whereas the Median Sweep (MS) method conducts cross-validation to estimate the TPR and the FPR for all possible thresholds on the training set. On the test set, it computes the prevalence estimate through the expression for the AC, for all thresholds where the corresponding values of TPR−FPR is equal to or greater than $1/4$, and returns the median of these as the final prevalence estimate. Forman evaluated the performance of the methods described above using a SVM as the base classification learner.

Barranquero (Barranquero et al., 2013) proposed proportion-weighted k-NN, called PWK$^{\alpha}$, by using weighted k-NN as the base learner for the AC method. Given a binary class problem represented by a collection of training sets $\{(\mathbf{x}_i, y_i), i = 1, ..., N\}$ and a test pattern $\mathbf{x}_j$, the estimated class value $\hat{y}_j$ is

Table 4.1: AC-based imbalance tolerant methods via threshold selection

Method	Training	Test
T50: TPR=50	Determine TPR and FPR for each potential decision threshold and set the decision threshold of classifier where TPR=50%	Same as AC
X: TPR crosses FPR	Same as T50, but set the threshold where (1−TPR)=FPR	Same as AC
MAX: TPR−FPR	Same as T50, but set the threshold where TPR−FPR is maximized	Same as AC
MS: Median Sweep	Determine TPR and FPR for each threshold t and record the values where TPR−FPR > 1/4	Median of AC estimates for all the recored thresholds)

obtained as

$$\hat{y}_j = sign(\sum_{i \sim j}^{k} y_i w_{y_i}^{(\alpha)}). \tag{4.7}$$

This approach uses the counts for all instances belong to each class on the training set to set the weight for the k-NN (S. Tan, 2005). The weight for the positive class is fixed to 1, and that for the negative class is computed as the adjusted quotient between the cardinalities of the negative class and those of the positive class, N_{-1} and N_{+1} respectively:

$$w_{-1}^{(\alpha)} = (\frac{N_{-1}}{N_{+1}})^{-1/\alpha} \; with \; \alpha \geq 1. \tag{4.8}$$

The algorithm described by Eq. (4.7) and Eq. (4.8) defines PWK$^{\alpha}$ and the PWK algorithm by fixing α=1. In the use of k-NN as a classification model, two

general approaches are used to assign weights: (1) assigning different weights to attributes, and (2) assigning different weights to each neighbor (Kang & Cho, 2008). In the latter approach, there are two types of weights: class weight and distance weight. PWK$^{\alpha}$ assigns different class weights to neighbors after calculating the distance. PWK$^{\alpha}$ provides flexibility such that the algorithm can be adapted to each dataset through the free parameter α, but requires an additional, expensive training procedure due to the set containing α. Moreover, experiments by Barranquero (Barranquero et al., 2013) showed no significant performance difference between PWK and PWK$^{\alpha}$.

4.3 Similarity-based adjusted count

There might be a difference in class distributions between the training and test sets because of the locality of data, which could have an effect on the TPR and the FPR. We are interested in this study in the basic assumption underlying the AC method, whereby there should be no fundamental change in the characteristics of the TPR and the FPR between the training and the test sets. Reliable estimation of the TPR and the FPR is crucial to the quality of the AC method. This task has been conducted through many-fold cross-validation in past research. In these contexts, the TPR and the FPR were obtained under the assumption that a priori class distribution $P(y)$ changes but the intra-class densities $P(\mathbf{x}/y)$ do not. However, if the minority instances of the test set are rare in such situations due to class imbalance and overlap, it can be difficult for the TPR and the FPR of the test set to retain the same regularities as

the training set. The TPR and the FPR could depend on the distribution, or the topology of the test patterns. Figure 4.1 shows that classifier performance varies with locality. The top part of the figure represents the TPR and FPR of satisfactory quality, whereas the bottom part does not.

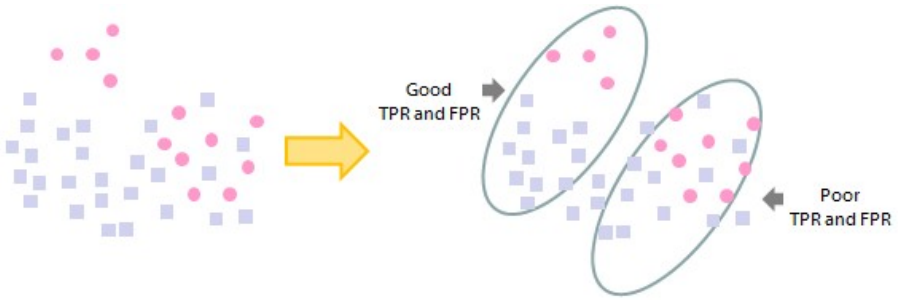


Figure 4.1: Differences in classifier performance by data location

The distribution and the topology of the test patterns need to be taken into account because they influence the TPR and the FPR. Classification performance yields identical characteristics for instances located close to one another. Thus, we can form a more accurate prevalence estimate by using training instances similar to test patterns. The similarity can be determined by a distance matrix on the feature space.

We propose similarity-based adjusted count (SAC), a simple and efficient prevalence estimation method that uses TPR and FPR for training instances similar to test instances. This method assumes that data distribution changes between the training and the test sets can affect classification performance, but similar instances in the feature space yield similar results. It uses training instances similar to test patterns, and thus retains adaptability to changes

in the test set. We use the k-NN classification model, which can remember the details of the data topology and search for similar instances based on distance. It searches training instances near the test patterns, called similar nearest neighbor (SNN), using k-NN. The SAC then estimates prevalence using class predictions for test patterns, and the TPR and the FPR for SNN.

The procedure involved in the SAC is shown in Figure 4.2. When the training set and the test patterns are given, it first computes class predictions for the training set and obtains $\hat{P}_{training}$, $\hat{N}_{training}$, and $p'_{training}$ for these. It then computes class predictions for test patterns, and checks $\hat{P}$ and $\hat{N}$ for p'. Following this, it searches the SNN for positive and negative predictions, called $SNN_{\hat{P}}$ and $SNN_{\hat{N}}$, respectively. $SNN_{\hat{P}}$ and $SNN_{\hat{N}}$ are searched for $\hat{P}_{training}$ and $\hat{N}_{training}$, respectively. SNN is the sum of $SNN_{\hat{P}}$ and $SNN_{\hat{N}}$. In nearest neighbor models, the nearest neighbors usually have identical class values. If some instances have a predicted class value different from that of their nearest neighbors, this would have been considered to have been caused by noise or anomalies. Thus, searching nearest neighbors within an identical predicted class value can remove the effect of noise. We search where the quotient between the cardinalities of the positive and negative predictions would be equal to that in the entire training set. This makes no difference in the predicted class skew between the SNN and the entire training set. For this, $SNN_{\hat{P}}$ and $SNN_{\hat{N}}$ should satisfy

$$\frac{\left|SNN_{\hat{P}}\right|}{\left|SNN_{\hat{N}}\right|} \approx \frac{p'_{training}}{1 - p'_{training}}. \tag{4.9}$$

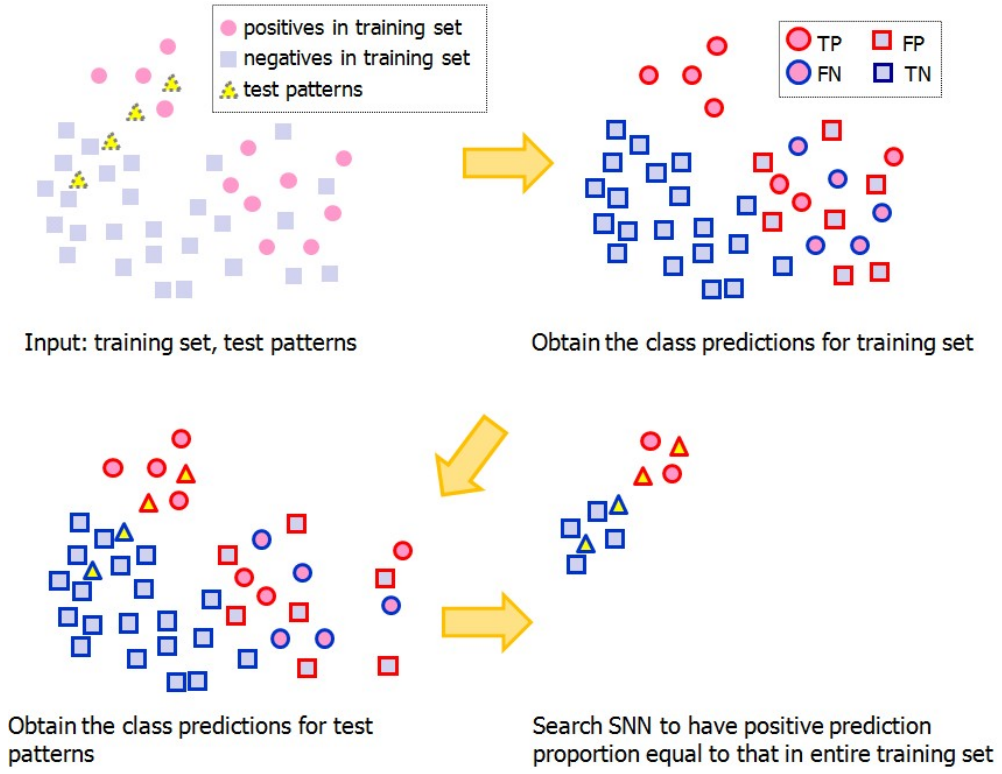


Figure 4.2: Procedure used by SAC

Thus, the parameter k for k-NN should be determined separately for each predicted class. We call each k for the positives and the negatives $k_{\hat{P}}$ and $k_{\hat{N}}$, respectively, for which appropriately values should be chosen to satisfy Eq. (4.9). In this study, we attempted to select SNN as scarcely as possible. If $p'/(1-p') \geq p'_{training}/(1-p'_{training})$, $k_{\hat{P}}$ is 1 and $k_{\hat{N}}$ is approximately $p'/(1-p') \times (1-p'_{training})/p'_{training}$. Otherwise, $k_{\hat{P}}$ is approximately $(1-p')/p' \times p'_{training}/(1-p'_{training})$ and $k_{\hat{N}}$ is 1. Finally, prevalence is estimated using the AC formula

with TPR and FPR only for SNN :

$$p'_{SAC} = \frac{p' - fpr_{SNN}}{tpr_{SNN} - fpr_{SNN}}. \tag{4.10}$$

The algorithm for the SAC is shown in Figure 4.3. The proposed SAC uses only part of the training set to estimate the TPR and the FPR, whereas the conventional AC method uses the entire training set for this. The number of positive predictions in the relevant part used by the SAC is proportionally equivalent to that of the entire training set. This constraint on the SAC allows us to compare the SAC and the AC under fair conditions using known class predictions. We can see class predictions but not true class labels. When p' is 0 or 1, the formulation of the SAC generates unfeasible parameter values, $k_{\hat{N}} = \infty$ or $k_{\hat{P}} = \infty$. In such cases, $k_{\hat{N}}$ and $k_{\hat{P}}$ are set to 1.

Inputs: $\{(\mathbf{x}_i, y_i), i = 1, ..., N\}$, the training set; $\{\mathbf{x}_j, j = 1, ..., N'\}$, the test patterns

Output: p'_{SAC}, the prevalence estimate using SAC

// Training phase:

Step 1: Determine weight for each class value c (+1,-1)

$w_{+1} = 1, w_{-1} = (N_{-1}/N_{+1})^{-1}$

Step 2: Obtain the estimated class for each training instance with PWK

For $i = 1 : N$, $\hat{y}_i = sign(\sum_{l\sim i}^{k} y_l w_{y_l})$

Step 3: Check $\hat{P}_{training}$, $\hat{N}_{training}$, $p'_{training}$

$\hat{P}_{training} \leftarrow \{(\mathbf{x}_i, y_i) : \hat{y}_i = 1\}$, $\hat{N}_{training} \leftarrow \{(\mathbf{x}_i, y_i) : \hat{y}_i = -1\}$

$p'_{training} \leftarrow \left|\hat{P}_{training}\right| / (\left|\hat{P}_{training}\right| + \left|\hat{N}_{training}\right|)$

// Test phase:

Step 4: Obtain the estimated class for each test pattern with PWK

For $j = 1 : N'$, $\hat{y}_j = sign(\sum_{i\sim j}^{k} y_i w_{y_i})$

Step 5: Check $\hat{P}$, $\hat{N}$, p' for test set

$\hat{P} \leftarrow \{\mathbf{x}_j : \hat{y}_j = 1\}$, $\hat{N} \leftarrow \{\mathbf{x}_j : \hat{y}_j = -1\}$, $p' \leftarrow \left|\hat{P}\right| / (\left|\hat{P}\right| + \left|\hat{N}\right|)$

Step 6: Set $k_{\hat{P}}$ and $k_{\hat{N}}$

If $p'/(1-p') \geq p'_{training}/(1-p'_{training})$,

$k_{\hat{P}} = 1$ and $k_{\hat{N}} = round(p'/(1-p') \times (1-p'_{training})/p'_{training})$

Else $k_{\hat{P}} = round((1-p')/p' \times p'_{training}/(1-p'_{training}))$ and $k_{\hat{N}} = 1$

Step 7: Search SNN for each test pattern among training set having same predicted class

$SNN \leftarrow \Phi$

For $j = 1 : N'$

If $\hat{y}_j$=1, $SNN_{\hat{P}} \leftarrow k_{\hat{P}}-NN$ *in* $\hat{P}_{training}$

Else $SNN \leftarrow SNN_{\hat{N}} \bigcup k_{\hat{N}}-NN$ *in* $\hat{N}_{training}$

$SNN \leftarrow SNN_{\hat{P}} \bigcup SNN_{\hat{N}}$

Step 8: Estimate the prevalence using AC method using TPR and FPR only for SNN

$TP_{SNN} \leftarrow \{(\mathbf{x}_i, y_i, \hat{y}_i) : y_i = 1 \wedge \hat{y}_i = 1, (\mathbf{x}_i, y_i) \in SNN\}$,

$FP_{SNN} \leftarrow \{(\mathbf{x}_i, y_i, \hat{y}_i) : y_i = -1 \wedge \hat{y}_i = 1, (\mathbf{x}_i, y_i) \in SNN\}$

$TPR_{SNN} \leftarrow |TP_{SNN}| / (SNN_{\hat{P}}), FPR_{SNN} \leftarrow |FP_{SNN}| / (SNN_{\hat{N}})$

$p'_{SAC} = \frac{p' - fpr_{SNN}}{tpr_{SNN} - fpr_{SNN}}$

Figure 4.3: Computation algorithm utilized by SAC

4.4 Performance on benchmark datasets

4.4.1 Experiment settings

We conducted experiments to verify the effectiveness of our proposed prevalence estimation method. The SAC method should estimate the prevalence of the positive class in a test set comparatively well, provided that we have a training set where the positive proportion is noticeably different from that in the test set. Therefore, the verification of our method required performance evaluation over a broad spectrum of test sets with different proportions of positive classes instead of a single test set. The minority class and the majority class of original dataset were designated the positive class and the negative class, respectively. To vary the test conditions, we followed the experimental methodologies used by (Forman, 2008) and Barranquero (Barranquero et al., 2013). This experiment employed same methodologies and benchmark datasets as are used in Chapter 2. It's because that this chapter and Chapter 2 have an identical study objective to accurately estimate the minority proportion in varying degrees of class imbalance and class overlap.

4.4.1.1 Prevalence estimation methods for comparison

The proposed SAC method was compared with AC-based methods. Prevalence estimation depends on the base classification learner. In this study, the conventional AC method was evaluated with both SVM and PWK as base learners. However, AC-based imbalance-tolerant methods (T50, X, MAX, and MS) were evaluated using only SVM, and the proposed SAC was tested only using

PWK. In case of the conventional AC, the TPR and the FPR were estimated by a 50-fold cross-validation using the training set. With regard to AC-based imbalance-tolerant methods and the SAC, the TPR and the FPR were estimated by immediately using the training set without dividing it for learning and testing. AC-based imbalance-tolerant methods estimated the TPR and the FPR by using the entire training set, whereas the SAC estimated the two measures for a few training instances similar to the test patterns. These methods were implemented offline using the commercial software package MATLAB (*Matlab version 7.10.0*, 2010). SVM was formulated in LIBSVM (Chang & Lin, 2011).

SVM models have a cost parameter C that controls the trade-off between allowing training errors and forcing rigid margins. Here, C was selected using ten-fold cross-validation and grid search from candidates [0.01 0.1 0 1 10]. PWK (S. Tan, 2005) models have a predefined parameter k to search for the NN. The value of k was selected using ten-fold cross-validation and grid search from candidates [3 5 7 9 11].

4.4.1.2 Evaluation measure

We measured absolute error (AE), which has been used to assess prevalence estimation methods in past studies (Barranquero et al., 2013; Forman, 2008). These studies proposed AE as the default loss function for prevalence estimation because it is very simple and intuitively interpretable:

$$AE = \left| p' - p \right| . \tag{4.11}$$

The Kullback-Leibler divergence has also been measured to assess the prevalence estimate in some contexts (Esuli & Sebastiani, 2010; Forman, 2008). How-

ever, KLD is inappropriate for this study because it is very sensitive to very small prevalence estimates in cases involving severe class imbalance. In such a situation, it is likely to yield undesirable results, such as indeterminate or even infinite values. A clear benefit of using AE is that it provides analysts with applicable information and actual error proportion within the range [0, 1].

4.4.2 Experiment results

We evaluated AE for seven methods on 87 binary class datasets (Lichman, 2013), applying stratified ten-fold cross-validation and varying the positive proportion of the test set at $\{p(\text{training}), 0, 0.1, \ldots, 0.9, 1.0\}$. This led to 870 training processes and 10,440 tests. The execution time of each experiment was not analyzed because it was heavily dominated by learning a classification model, and was hardly affected by the post-processing approach of prevalence estimation itself.

4.4.2.1 Overview analysis

For each binary class dataset, the AE was calculated for 12 test conditions. Appendix C shows the AE distribution according to test condition for each dataset. We averaged the AE from 12 repetitions (12 test conditions) for each binary-class dataset. The averaged AE of each prevalence estimation method for each binary-class datasets is shown in Table 4.2. The SAC with PWK exhibited the best performance, followed by the AC with PWK, for most datasets. The difference between the values yielded by these methods ranged from -0.010 to 0.022.

Table 4.2: AE for each binary-class dataset

Identifier	SVM					PWK	
	AC	T50	X	MAX	MS	AC	SAC
Haberman	0.458	0.186	0.218	0.219	0.183	0.330	0.293
Iris.1	0.000	0.226	0.021	0.021	0.083	0.015	0.000
Iris.2	0.275	0.312	0.237	0.237	0.227	0.067	0.047
Iris.3	0.060	0.178	0.051	0.050	0.061	0.064	0.043
Blood	0.479	0.122	0.162	0.158	0.155	0.185	0.206
Balance.1	0.464	0.507	0.503	0.506	0.506	0.320	0.428
Balance.2	0.074	0.166	0.077	0.071	0.095	0.111	0.109
Balance.3	0.038	0.160	0.047	0.041	0.084	0.125	0.117
Liver	0.196	0.164	0.192	0.191	0.178	0.242	0.187
Seeds.1	0.065	0.124	0.068	0.053	0.058	0.065	0.072
Seeds.2	0.019	0.174	0.034	0.034	0.071	0.053	0.033
Seeds.3	0.051	0.131	0.051	0.053	0.061	0.056	0.033
Ecoli.1	0.032	0.077	0.030	0.027	0.030	0.025	0.03
Ecoli.2	0.126	0.142	0.081	0.080	0.063	0.081	0.069
Ecoli.3	0.204	0.199	0.117	0.097	0.111	0.097	0.101
Ecoli.4	0.058	0.202	0.071	0.073	0.116	0.092	0.077
Ecoli.5	0.121	0.182	0.076	0.080	0.090	0.068	0.081
Pima	0.074	0.084	0.076	0.077	0.060	0.087	0.110
Breast.1	0.134	0.192	0.139	0.145	0.114	0.132	0.101
Breast.2	0.224	0.230	0.200	0.164	0.173	0.169	0.139
Breast.3	0.502	0.401	0.381	0.346	0.322	0.275	0.278
Breast.4	0.222	0.261	0.140	0.144	0.174	0.121	0.123
Breast.5	0.135	0.284	0.076	0.122	0.138	0.092	0.064
Breast.6	0.062	0.104	0.083	0.083	0.093	0.053	0.020
Wisconsin	0.027	0.114	0.026	0.017	0.032	0.020	0.024
Tic-Tac-Toe	0.487	0.262	0.276	0.216	0.216	0.053	0.133
Contraceptive.1	0.211	0.127	0.124	0.125	0.116	0.491	0.152
Contraceptive.2	0.477	0.219	0.274	0.255	0.253	0.385	0.187
Contraceptive.3	0.487	0.328	0.304	0.247	0.247	0.487	0.239
Glass.1	0.013	0.081	0.034	0.034	0.047	0.025	0.014
Glass.2	0.175	0.204	0.169	0.167	0.156	0.099	0.062
Glass.3	0.049	0.206	0.123	0.123	0.092	0.046	0.055
Glass.4	0.251	0.401	0.163	0.165	0.199	0.092	0.080
Glass.5	0.049	0.201	0.080	0.080	0.068	0.073	0.065

Continued

Table 4.2 –continued from previous page

Identifier	SVM					PWK	
	AC	T50	X	MAX	MS	AC	SAC
Vowel.1	0.466	0.159	0.121	0.100	0.090	0.004	0.003
Vowel.2	0.466	0.199	0.104	0.098	0.065	0.004	0.001
Vowel.3	0.107	0.113	0.056	0.064	0.060	0.006	0.002
Vowel.4	0.466	0.213	0.125	0.131	0.105	0.019	0.015
Vowel.5	0.466	0.279	0.30	0.215	0.215	0.025	0.021
Vowel.6	0.466	0.308	0.278	0.267	0.265	0.014	0.011
Vowel.7	0.120	0.139	0.058	0.058	0.071	0.007	0.003
Vowel.8	0.466	0.330	0.326	0.247	0.225	0.015	0.010
Vowel.9	0.107	0.170	0.036	0.037	0.052	0.007	0.005
Vowel.10	0.466	0.438	0.496	0.301	0.301	0.017	0.011
Vowel.11	0.067	0.122	0.042	0.057	0.052	0.001	0.000
Wine.1	0.009	0.112	0.032	0.032	0.063	0.039	0.020
Wine.2	0.035	0.161	0.058	0.058	0.054	0.036	0.017
Wine.3	0.036	0.207	0.058	0.058	0.082	0.035	0.023
Statlog.1	0.036	0.080	0.027	0.027	0.035	0.028	0.031
Statlog.2	0.078	0.094	0.079	0.079	0.056	0.062	0.079
Statlog.3	0.033	0.057	0.023	0.028	0.032	0.024	0.020
Statlog.4	0.159	0.137	0.077	0.080	0.063	0.082	0.087
Image.1	0.007	0.067	0.003	0.003	0.032	0.004	0.004
Image.2	0.000	0.069	0.004	0.004	0.025	0.000	0.000
Image.3	0.028	0.048	0.02	0.020	0.031	0.012	0.015
Image.4	0.061	0.084	0.041	0.042	0.031	0.014	0.019
Image.5	0.052	0.065	0.038	0.037	0.044	0.021	0.017
Image.6	0.006	0.049	0.008	0.008	0.026	0.006	0.006
Image.7	0.003	0.068	0.003	0.003	0.028	0.004	0.003
Cardiotocography.1	0.050	0.172	0.042	0.045	0.109	0.049	0.041
Cardiotocography.2	0.060	0.138	0.045	0.047	0.108	0.080	0.071
Cardiotocography.3	0.131	0.209	0.096	0.095	0.109	0.110	0.101
Parkinsons	0.116	0.130	0.157	0.134	0.112	0.091	0.061
Ionosphere	0.110	0.116	0.084	0.111	0.099	0.073	0.091
SPECTF heart	0.204	0.236	0.168	0.151	0.164	0.235	0.215
Sonar	0.147	0.218	0.151	0.151	0.136	0.103	0.075
Digits.1	0.017	0.097	0.026	0.026	0.028	0.000	0.000
Digits.2	0.034	0.134	0.032	0.032	0.064	0.011	0.011
Digits.3	0.015	0.136	0.048	0.048	0.044	0.018	0.014
Digits.4	0.040	0.090	0.035	0.033	0.037	0.013	0.011

Continued

Table 4.2 –continued from previous page

Identifier	SVM					PWK	
	AC	T50	X	MAX	MS	AC	SAC
Digits.5	0.037	0.127	0.051	0.051	0.047	0.014	0.014
Digits.6	0.028	0.087	0.028	0.034	0.038	0.015	0.014
Digits.7	0.020	0.078	0.032	0.032	0.040	0.003	0.002
Digits.8	0.036	0.155	0.053	0.053	0.047	0.008	0.007
Digits.9	0.061	0.093	0.045	0.051	0.044	0.018	0.017
Digits.10	0.036	0.090	0.032	0.035	0.044	0.015	0.015
Hill-Valley	0.052	0.048	0.012	0.012	0.016	0.162	0.194
Urban.1	0.162	0.131	0.214	0.214	0.069	0.056	0.037
Urban.2	0.122	0.100	0.155	0.155	0.087	0.059	0.058
Urban.3	0.052	0.188	0.155	0.155	0.114	0.058	0.058
Urban.4	0.139	0.090	0.173	0.173	0.071	0.069	0.065
Urban.5	0.169	0.157	0.171	0.179	0.148	0.053	0.050
Urban.6	0.080	0.166	0.181	0.181	0.091	0.060	0.060
Urban.7	0.064	0.117	0.152	0.152	0.082	0.043	0.035
Urban.8	0.153	0.172	0.26	0.260	0.116	0.123	0.126
Urban.9	0.085	0.112	0.159	0.159	0.065	0.044	0.041
Musk	0.094	0.133	0.131	0.131	0.066	0.058	0.058

Table 4.3 shows the summary of the AE values for each prevalence estimation method over all binary class datasets. The average AE was first calculated for each prevalence estimation method for all datasets. For each method, we then counted the number of cases where the SAC had recorded the best AE across all datasets. The SAC outperformed all other methods in 49 of the 87 datasets.

Figure 4.4 shows the AE results by box-plot for all 12 test conditions in all 87 datasets. We can observe the range of AE scores for every method. Taking into account the main elements of the box-plot, we could check if the SAC stood out as the most compact system in terms of inter-quartile range. The SAC

Table 4.3: Summary of AE for total test conditions

Measure	SVM					PWK	
	AC	T50	X	MAX	MS	AC	SAC
Average	0.147	0.166	0.115	0.109	0.103	0.078	0.066
N. best datasets	7	2	2	7	7	13	49

recorded its first quintile, its median, and its third quintile around 0.011, 0.035, and 0.087, respectively. Each of them was the lowest value of all prevalence estimation methods. Thus, the SAC yielded the most competitive prevalence estimates across al1 12 test conditions, the target prevalence of which varied from 0% to 100%.

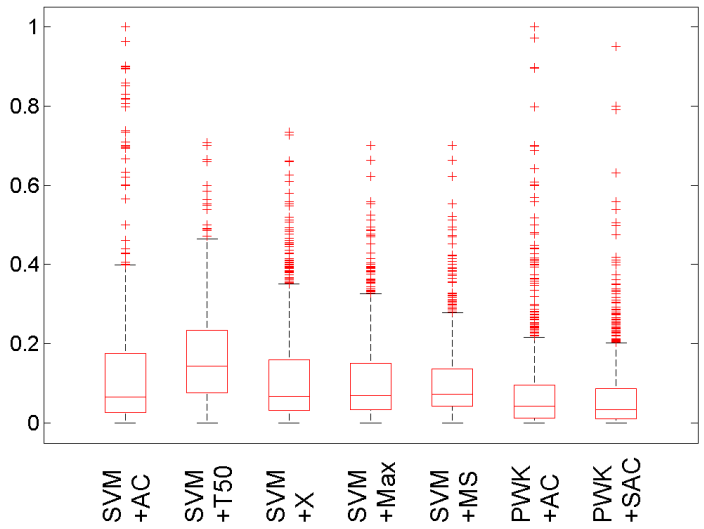


Figure 4.4: Box-plot of AE

We also carried out a two-step statistical test, following Demšar's proposal (Demšar, 2006) that the Friedman test (M. Friedman, 1940) with the corresponding post-hoc tests be used to compare of performance with multiple algorithms over multiple datasets. The first step consisted of a Friedman test of the null hypothesis that all prevalence estimation methods performed equally in terms of AE rank. They obtained the rank of every method for different test conditions, and computed an aggregated rank per dataset. In this study, these statistical tests compared seven prevalence estimation methods over 87 binary class datasets with 12 repetitions.

Friedman's null hypothesis was rejected at with a 5% significance level. The results of the post-hoc Nemenyi test (Nemenyi, 1963) are shown in , where prevalence estimation methods were sorted by average rank in ascending order. In the Nemenyi test, the null hypothesis (no performance difference between the two algorithms) was rejected if the difference between the average ranks was greater than the critical distance. The value of CD for the Nemenyi test on 87 datasets for seven methods was 0.966 at α=0.05. There was no evidence of significant differences among prevalence estimation methods joined by the horizontal lines in Figure 4.5. Therefore, the figure implies that the SAC performed significantly better than the other methods in terms of AE.

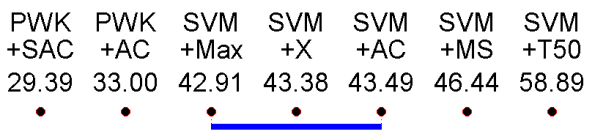


Figure 4.5: Nemenyi post-hoc tests for total test conditions at α=0.05

4.4.2.2 Analysis of results by target prevalence

Although the foregoing account provided interesting evidence of the quality of our proposed SAC method, they fail to show the effect of positive class prevalence in the test set in a detailed manner. They simply offer an aggregated overview of the distributions and the ranges of AE without taking into account the error distribution with regard to varying target prevalence. We examine here the performance of each prevalence estimation method with respect to changes in target prevalence. In Appendix C, the vertical gray line represents the positive class prevalence of the training set. We can check that all datasets resulted in a similar pattern. The SAC represented overall satisfactory performance across varying test prevalence for every dataset. The estimated prevalence of the SAC was not far removed from actual prevalence regardless of the target value in comparison with other methods.

Table 4.4 shows the summary of AE values per test condition over all datasets. The SAC exhibited the best performance across nine different prevalence conditions of the test set. In cases of extreme target prevalence (0%, 10%, and 100%), the SAC was second best. When using PWK as the classification learner, the SAC performed better than the AC for all 12 test conditions in terms of the number of datasets.

The Friedman test with the corresponding post-hoc tests was conducted for a comparison of AE with the seven prevalence estimation methods per prevalence over 87 datasets with no repetition. Friedman's null hypothesis was rejected at α=0.05 for all instances of test prevalence. The values of the CD for

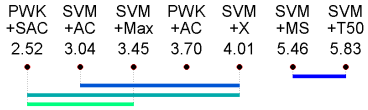
Table 4.4: AE per test condition

% positives	Measure	SVM					PWK	
		AC	T50	X	MAX	MS	AC	SAC
p(training)	Average	0.059	0.101	0.078	0.064	0.086	0.062	0.040
	N. best datasets	25	1	8	15	2	5	35
0	Average	0.030	0.055	0.062	0.053	0.060	0.025	0.035
	N. best datasets	19	37	4	10	2	8	21
10	Average	0.05	0.092	0.077	0.063	0.088	0.050	0.050
	N. best datasets	29	3	7	12	4	12	27
20	Average	0.072	0.119	0.091	0.077	0.102	0.062	0.050
	N. best datasets	26	0	7	11	3	10	34
30	Average	0.097	0.147	0.102	0.087	0.109	0.072	0.054
	N. best datasets	19	0	5	12	3	11	38
40	Average	0.121	0.167	0.111	0.100	0.115	0.080	0.061
	N. best datasets	13	0	5	10	5	15	39
50	Average	0.152	0.198	0.118	0.118	0.126	0.088	0.071
	N. best datasets	10	1	7	8	5	12	44
60	Average	0.182	0.220	0.135	0.131	0.132	0.094	0.081
	N. best datasets	8	3	6	5	7	14	44
70	Average	0.21	0.230	0.144	0.142	0.129	0.097	0.086
	N. best datasets	4	3	6	7	7	17	43
80	Average	0.246	0.234	0.150	0.150	0.113	0.105	0.092
	N. best datasets	4	3	4	3	14	14	45
90	Average	0.265	0.224	0.159	0.161	0.105	0.107	0.088
	N. best datasets	3	5	2	3	17	9	49
100	Average	0.284	0.205	0.153	0.158	0.072	0.099	0.082
	N. best datasets	1	4	3	0	42	11	42

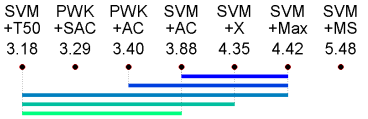
the Nemenyi test with 87 datasets and seven methods was 0.966 at α=0.05, which was identical to that in terms of the aggregated prevalence analysis in Section 5.2 because it was not dependent on repetition. The overall results of the Nemenyi test are shown in Figure 4.6. In case the positive prevalence of the test set was 0%, 10%, and 100%, although the performance of the SAC

was slightly worse than the first runner, there was no statistically significant difference. The SAC recorded the best performance in the other nine test conditions, and this superiority was sometimes statistically significant. This leads us to conclude that the SAC is the most suited to deal with a variety of datasets and test conditions.

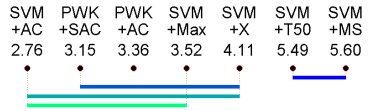
Additionally, We applied the proposed SAC method into the MPI dataset to predict prevalence and conducted further analyses in Appendix D.



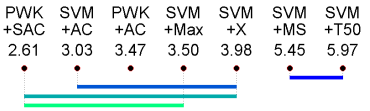
(a) p(test)



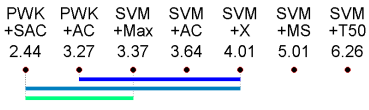
(b) 0%



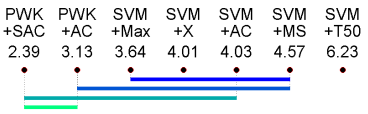
(c) 10%



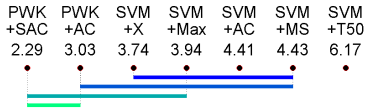
(d) 20%



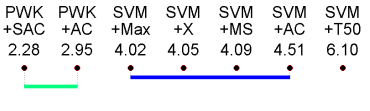
(e) 30%



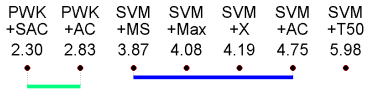
(f) 40%



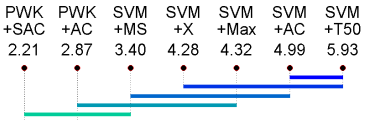
(g) 50%



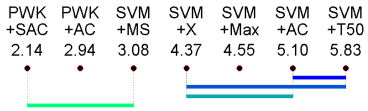
(h) 60%



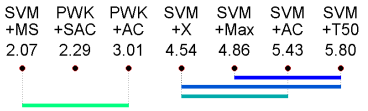
(i) 70%



(j) 80%



(k) 90%



(l) 100%

Figure 4.6: Nemenyi post-hoc tests per test condition α=0.05

4.5 Summary

In this study, we proposed the SAC, is an adaptive approach to estimate prevalence by changing the proportion of total positives. Estimation of prevalence can help cope with class imbalance and class overlap problems where direct application of the outcomes of classification models may result in substantial error. The SAC can accurately estimate prevalence even when the test and training sets have different data distributions. This method is based on the AC technique, and uses the TPR and the FPR values of training instances that are similar to test instances. The SAC method efficiently uses a part of the entire training set to estimate the TPR and the FPR, whereas the conventional AC uses the entire training set.

In order to verify the effectiveness of the SAC, we conducted experiments with 87 binary class datasets with changing positive class prevalence in the test set. We tested the SAC and other AC-base quantification methods (AC, T50, X, MAX, and MS) using SVM or PWK as base classification learner in terms of AE. The proposed SAC was the outstanding method in terms of AE across all binary class datasets. The superiority of the SAC was decisive in terms of average, the number of datasets that yielded the best performance of the 87 binary class datasets, and average rank. Further, the SAC exhibited satisfactory performance across different test prevalence values ranging from 0% to 100% for each dataset. The SAC recorded the best results in nine test conditions, and the superiorities were sometimes statistically significant. In case of extreme target prevalence (0%, 10%, 100%), the SAC was second-best, but

was not statistically significantly difference from the best method. When using PWK as classification learner, the SAC performed better than the AC for all 12 test conditions. It estimated the prevalence comparatively well regardless of the changing proportion of the positive class, and this was the key objective of this study.

The promising experimental results have encouraged us to apply our method to real-world problems in future research. Our proposed method can be applied to personnel management in fields that require careful recruiting and attention, such as the military. For example, estimating maladjustment probability is very important in predicting the management of military personnel. Prevalence estimates can be used to predict maladjustment proportion among different corps/division. If the relevant ministry identifies a division with a higher prevalence of maladjustment, it should instruct the commander of the division to actively prevent the maladjusted problem (e.g., counsel, mentoring, education), and reassign some of the maladjusted soldiers to another division. For another example, estimating fault rate is useful in predicting the quality management of product lines. A manufacturing company can estimate the annual number of items that will be defective in production lines without having to identify particular faulty products. Production lines with greater fault rates should have their products inspected more thoroughly. Furthermore, medical diagnosis, text categorization, and risk/disease prediction require prevalence estimation because it supports decision making based on the relative and absolute priorities in terms of cost effectiveness. Prevalence estimates can also be used as prior information for classification tasks. Bayesian classification mod-

els require the parameter of class prior in order to determine class predictions. Thus, the prevalence estimate can be applied as the class prior value of a Bayes classifier.

Chapter 5

Application of prevalence estimate into Bayesian classifier

5.1 Background

In this chapter, the prevalence estimate is applied as class prior for obtaining calibrated probability or classifying instances from Bayes classifiers. Bayes classifier can be represented as Bayesian network among attributes and class. Bayesian classification algorithms allow us to express the posterior probability in terms of the evidence, class conditional probability, and prior probability for each class value. Thus prior probability for a class value directly affects classification decision. The class prior is intrinsically identical to prevalence in this study. The use of prevalence estimate as class prior knowledge is expected to generate more accurate calibrated probabilities and class predictions than other approaches that rely on a class prevalence of training set, provided that we have a training set where the positive proportion may be noticeably different from that in test set.

Additionally, we propose a hybrid Bayesian classification algorithm, namely correlation-based Gaussian Bayesian network (CGBN), which considers both

classification accuracy and intrinsic dependence between attributes. CGBN has a BN structure of making attribute clusters and is a type of semi-naïve Bayesian classifier that fixes a cluster structure from the undirected graphical model. CGBN is expected to provide more accurate classification performance than the previously introduced Bayes classification algorithms.

A Bayesian network (BN) provides a graphical representation of the probabilistic relationships among a set of attributes (S. Lauritzen & Spiegelhalter, 1988; N. Friedman et al., 1997; Neapolitan, 2003; Pearl, 1988). A BN is defined by a pair B=(G, Θ), where G is a graphical structure whose nodes, X_1, X_2, ..., X_d, represent attributes, and whose edges represent direct dependencies among the attributes. The second component, Θ, denotes a set of parameters of the network. Bayesian classification algorithms allow us to express posterior probability in terms of the evidence, class conditional probability, and prior probability for each class value c as:

$$p(c/x_1, x_2, ..., x_d) = \frac{p(x_1, x_2, ..., x_d/c)p(c)}{p(x_1, x_2, ..., x_d)}. \tag{5.1}$$

The BN, as a Bayesian classifier (Duda & Hart, 1973), combines the posterior probability of class $p(Y/X_1,X_2,..,X_d)$, with a decision rule for predicting class value. A common rule is to pick the class value that is most probable, as in Eq. (5.2); This is known as the maximum a-posteriori decision rule.

$$C^{Bayes}(x_1, x_2, ..., x_d) = \arg\max_c \frac{p(x_1, x_2, ..., x_d/c)p(c)}{p(x_1, x_2, ..., x_d)}. \tag{5.2}$$

Belonging to the family of probabilistic graphical models (PGMs), BNs combine principles from graph theory, probability theory, computer science, and

statistics (Ben-Gal, 2007). PGMs with undirected edges are generally called Markov networks, and other those with directed edges are known as directed acyclic graphs (DAGs). BNs, corresponding in general to DAGs (Auvray, 2007; Bøttcher, 2004), can be constructed by finding Bayesian factors among attributes (Bøttcher, 2004; Geiger & Heckerman, 1994; Giudici & Green, 1999; Heckerman, 1995). A DAG is usually transformed into a decomposable Markov network for probability inference (Heckerman et al., 1995). During this transformation, two graphical operations—namely, moralization and triangulation—are performed on the DAG (Jayech & Mahjoub, 2011).

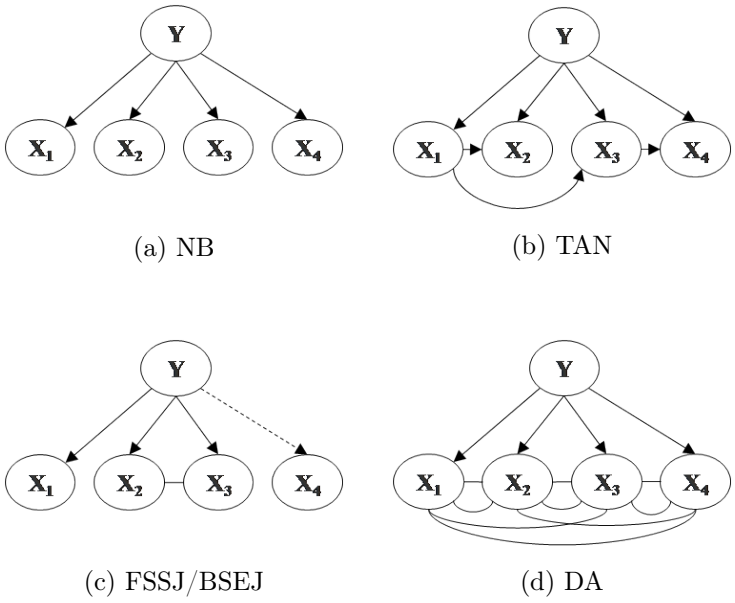
The remainder of this chapter is structured as follows. In Section 5.2, we briefly review previous work related to various GBNs that are applied to data with numerical attributes. In Section 5.3, we explain the proposed methods for finding bCGBN and bCGBN-BSE, along with the method for reducing computational complexity. In Section 5.4, we outline our experimental settings for evaluating the proposed methods and analyze the experimental results. In Section 5.5, we conclude by discussing the contributions made in this study and look at possible future work.

5.2 Related work

5.2.1 BN learning from the undirected graphical model

A BN specifies probabilistic relationships among a set of attributes using graphical representation. In principle, there are exponentially many BNs that can be constructed from a given set of attributes. The well-known Naïve Bayes

(NB) can be regarded as a kind of BN (Kononenko, 1990), assuming the class-conditional independence of all of attributes, as shown in Figure 5.1. (a).



(a) NB (b) TAN

(c) FSSJ/BSEJ (d) DA

Figure 5.1: Example of different BNs

If the number of attributes is large, it is computationally unfeasible to calculate all possible BNs that correspond to a DAG. Robinson (Robinson, 1973) and Stanley (Stanley, 1973) found that the number of possible DAGs that contain d nodes, f(d), can be counted by using the following recursive formula:

$$f(d) = \sum_{i=1}^{d} (-1)^{i-1} \begin{pmatrix} d \\ i \end{pmatrix} 2^{i(d-i)} f(d-i). \tag{5.3}$$

The formula can be proved using the inclusion–exclusion (Liskovets, 1975) method with respect to the property of a given DAG containing (d−i) nodes, where i spans over 1, . . . , d. We count triples (S_i, D_i, E_i) at each ith iteration,

where S_i is the subset with i elements in [1, ..., d], D_i are the DAGs on [1, ..., d]$-S_i$, and E_i are the DAGs of the set $S_i \times$ ([1, ..., d]$-S_i$) (Gessel, 1996). We consider the elements of E to be edges from S to D. Between any two nodes (one belonging to S and the other belonging to D), either no connection can be generated or a directed edge from S to D can. Obviously, the number of DAGs of a subset in S_i is equal to $2^{i(d-i)}f(d-i)$ with redundancy. Through the iterations, a large number of DAGs with n nodes are counted and redundant DAGs are eliminated. The number of DAGs on n attributes, for d=1, 2, 3 is 1, 3, 25, 543, 29281, 3781503, etc., as shown in Figure 5.2. If d=3, the total number of DAGs is 25 because the number of E_1, E_2, and E_3 is 36, 12, and 1, respectively. Unfortunately, since the number of possible DAGs involving n attributes yields a complexity of $2^{O(d^2 \log d)}$ for large numbers of d, such an exhaustive approach is impractical (Auvray, 2007; N. Friedman & Koller, 2003).

Identifying the edges and the corresponding probability instantiations of all attributes in the general inference cases for BNs have been shown to be NP-hard (Chickering et al., 1994; Cooper, 1990; Shimony, 1994; Hambrush & Tu, 1997; S. Li, 1995). As a consequence, heuristic algorithms have become the standard methodology for addressing BN learning (Dagum & Luby, 1993). BNs corresponding in general to a DAG (Auvray, 2007; Bøttcher, 2004), can be constructed by finding Bayesian factors among attributes (Giudici & Green, 1999; Heckerman et al., 1995). These strategies use Bayesian factors to compare BNs that differ by the direction of a single arrow. However, some heuristic alternatives involve fixing a unique structure from the undirected graphical model and learning the network structure from a non-Bayesian point of view.

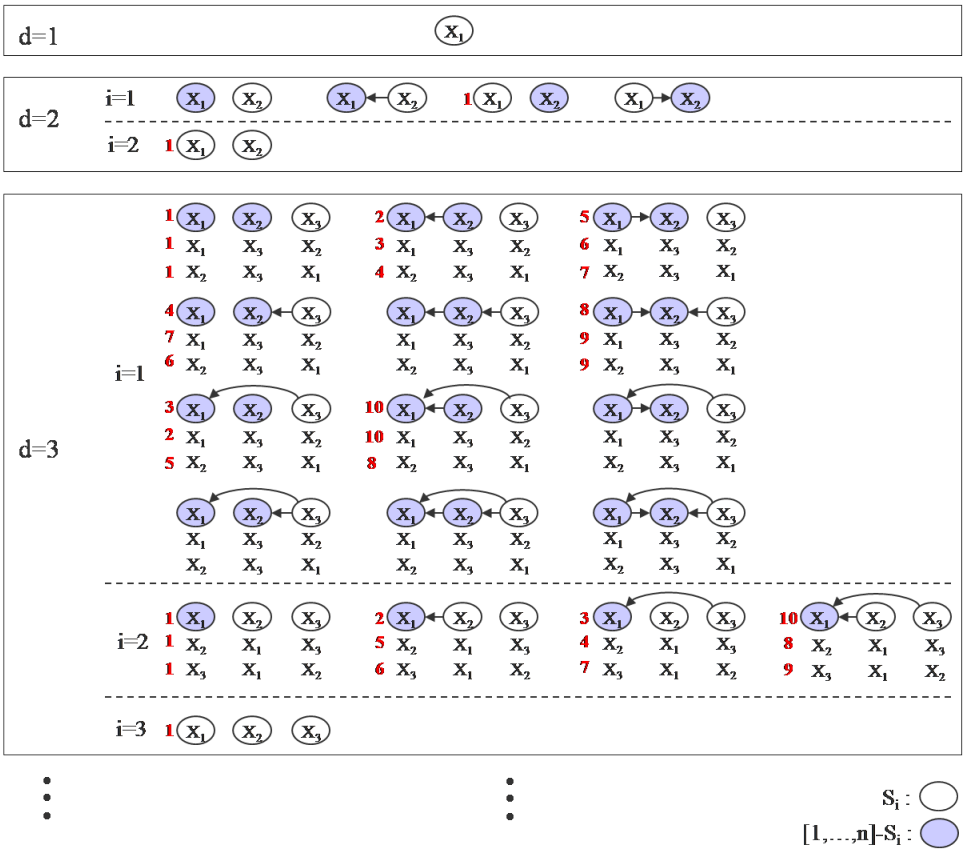


Figure 5.2: DAGs with d nodes at iteration i in Eq. (5.3) Red numbers denote redundant DAG identifiers at each d

The undirected models permit the inclusion of a larger number of connections among attributes because they are no longer restricted by the decomposability requirements imposed by the chain rule in DAGs (Jakulin & Irish, 2006).

Chow and Liu (Chow & Liu, 1968) introduced the tree augmented naïve (TAN) Bayes, a kind of BN representing a maximum spanning tree structure except for the class variable, as as shown in Figure 5.1(b). TAN is defined by the constraint that each attribute has a class variable as a parent, and all at-

tributes except one (root attribute) should have another attribute as parent. The maximum spanning tree can be searched using a greedy method, such as Prim's algorithm (Prim, 1957) whose complexity is $O(d^2)$. TAN uses mutual information or class-conditional mutual information of attributes as an arc value (N. Friedman et al., 1997; Pérez et al., 2006). Recent studies (N. Friedman et al., 1997; Pérez et al., 2006) have preferred class-conditional mutual information to mutual information. The class-conditional probability can be computed as the product of the probability of each attribute, given its parent attribute and the relevant class variable (Mitchell, 1997). When using the Gaussian distribution function, the class-conditional probability for an instance $(x_1, ..., x_d)$ is estimated as follows:

$$P(x_1, ..., x_d/c) = P(x_r/c) \prod_{i \neq r} P(x_i/(x_{parent(i)}, c)) \tag{5.4}$$

where x_r denotes a root attribute, and $x_{parent(i)}$ denotes a parent attribute of x_i.

Moreover,

$$x_r/c \sim N(\mu_{x_i/c}, \sigma^2_{x_i/c}), \tag{5.5}$$

and

$$x_i/x_{parent(i)} \sim N(\mu_{x_i} + \frac{\sigma_{x_{parent(i)}}}{\sigma_{x_i}} \rho(x_{parent(i)} - \mu_{x_{parent(i)}}), (1-\rho^2)\sigma^2_{x_i}) \tag{5.6}$$

where ρ denotes r between X_i and $X_{parent(i)}$.

TAN has some limitations as a classifier. First, the structural likelihood maximization that considers mutual information does not imply the maximization of accuracy for classification. Second, TAN is a complete spanning tree structure, due to which some irrelevant arcs may be added.

Moreover, some search strategies can neglect Bayesian factors in both training and probability estimation. Pazzani (Pazzani, 1997) devised a BN with a cluster structure, where the attributes within a single cluster were assumed to be correlated and those belonging to different clusters were assumed to be independent of one another. He proposed a forward sequential selection and joining (FSSJ) algorithm and a backward sequential elimination and joining (BSEJ) algorithm, as shown in Figure 5.1(c). These algorithms employ the kinds of greedy search processes that sequentially form clusters. FSSJ starts with a Bayesian classifier of an empty attribute set that simply classifies all instances to the most frequent class value. The following two operators are then used for each attribute to generate new classifiers:

Step 1: Add a new attribute independently to the current classifier

Step 2: Join a new attribute to an attribute group used by the current classifier

On the other hand, BSEJ initially creates a BN by treating all attributes as class-conditionally independent. The following two operators are then used for each attribute to generate new classifiers:

Step 1: Remove an attribute used by the current classifier

Step 2: Join an attribute in a cluster, used by a given classifier, with another used by the same classifier

In FSSJ and BSEJ, every classifier is tested in terms of classification accuracy and the best classifier is chosen. If no classifier yields an improvement over

a given classifier, the algorithm is terminated and the given classifier is returned as the chosen one. FSSJ and BSEJ do not only make attribute clusters, but also select attributes sets for classification. In FSSJ and BSEJ, class-conditional probability can be estimated using the following Gaussian distribution:

$$P(x_1, ..., x_d/c) = \prod_{m=1}^{M} P(g_m/c) \tag{5.7}$$

where g_m denotes an attribute set belonging to the mth cluster group. Moreover,

$$P(g_m/c) \sim N(\mu_{g_m/c}, \Sigma_{g_m/c}). \tag{5.8}$$

This is a kind of semi-naïve Bayesian classifier to restrict a cluster structure from the undirected graph, assuming a conditional dependency exists among the clustered attributes (Huang et al., 2003; Liang & Srebro, 2004).
where $\mu_{g_m/c}$ is the mean matrix of g$_m$ conditioned to the class value Y=c, and $\Sigma_{g_m/c}$ is the covariance matrix of g$_m$ conditioned to the class value Y=c. $\mu_{g_m/c}$ and $\Sigma_{g_m/c}$ can be estimated based on the sample mean and sample variance of all training instances belonging to Y=c.

Pazzani (Pazzani, 1997) initially introduced FSSJ and BSEJ for categorical attributes, and Pérez (Pérez et al., 2006) applied these algorithms to numerical attributes with a Gaussian probability function. In these studies, BSEJ was more accurate than FSSJ for datasets resembling real-world configurations. Using FSSJ and BSEJ, several attributes can be joined by an iterative application of the joining operator, and $O(d^3)$ Bayesian classifiers can be tested in the worst case. These algorithms consider all one-step modifications of a classifier at any given time, as well as modifications of classifiers that yield high computational

complexity. Although they can join more than two attributes, this process must be carried out in multiple steps.

Bayesian consideration for classification with discriminant analysis (DA) (Çalış & Erol, 2012; Chen et al., 2004; Cooley, 1975) is a kind of Bayesian classifier that assumes dependence among all attributes, as shown in Figure 5.1(d), and uses a multivariate Gaussian distribution function for all attributes X_1, X_2, ..., X_d. Linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) are widely used techniques, and involve different assumptions for covariance as follows. LDA (Fisher, 1936) is used when data for every class value is assumed to share a common covariance. Unlike LDA, however, in QDA (Johnson & Wichern, 2002; Mclachlan, 2004) there is no assumption whereby the covariance of all classes is identical. Hence, in QDA, the covariance matrix is estimated separately for each class value. QDA has better data approximation and predictability power in classification problems than LDA. Furthermore, Pérez et al. showed that the accuracy of Bayesian quadratic discriminant analysis (BQDA) can be improved through backward sequential elimination (BSE).

5.2.2 Filter and wrapper approaches

Identifying probabilistic relationships among attributes in BN learning is a Nondeterministic Polynomial-time (NP)-hard problem. In principle, we can take into account all possible DAGs and select the one that exhibits the best performance. Unfortunately, since the number of possible DAGs involving d attributes incurs a complexity of $2^{O(d^2 \log d)}$ for large values of d, such an approach is impractical (Auvray, 2007; N. Friedman & Koller, 2003). BNs have been shown to

be NP-hard with regard to locating edges and the corresponding probability instantiations of all attributes in general inference cases (Cooper, 1990; Garey & Johnson, 1979; S. Li, 1995; Shimony, 1994; Hambrush & Tu, 1997). Therefore, heuristic algorithms have become the standard methodology for addressing BN learning (Teyssier & Koller, 2005; Scutari & Brogini, 2012).

Although most previous work on BNs focused on DAGs, some heuristic alternatives involve fixing a structure, such as a tree or a cluster, in the undirected graphical model, and learning the network structure from a non-Bayesian point of view. Such undirected models permit the inclusion of a larger number of connections among attributes because they are no longer restricted by the decomposability requirements imposed by the chain rule in DAGs. Learning strategies for the undirected models are guided by evaluation criteria such as likelihood or accuracy (Shimony, 1994). Depending on the nature of these criteria, we think that structural learning can be carried out in two ways: a filter approach that uses likelihood to identify intrinsic relationships among attributes, and a wrapper approach that uses estimated predictive accuracy to conduct a search. Filter approaches in general incur short computation times in training, but do not guarantee the maximization of predictive accuracy (N. Friedman et al., 1997; Grossman & Domingos, 2004; Sahami, 1996; Langley & Sage, 1994). Wrapper approaches principally find the most useful structure, but practically incur high computation cost and cause over-fitting by overusing predictive accuracy (Guyon & Elisseeff, 2003; Kohavi & Sommerfield, 1995; Talavera, 2005).

5.2.3 Covariance constraint for Gaussian distribution

A Gaussian Bayesian network (GBN), i.e., a BN that uses a Gaussian distribution, was usually chosen to approximate the class-conditional probability for a numerical attribute in previous studies (Gómez-Villegas et al., 2007; S. L. Lauritzen & Wermuth, 1989; Pérez et al., 2006; Bøttcher, 2004). The Gaussian distribution is a continuous probability distribution that is often used as a first approximation to describe real-valued numerical attributes that tend to be densely populated around a single mean value. The Gaussian distribution is a symmetric bell-shaped function that has maximum likelihood at a single mean value and is very effective for an approximation of the distribution of numerical attributes in the real world (Hermstein & Murray, 1994). The distribution is characterized by two parameters: a mean matrix (μ) and a covariance matrix (Σ) for all attributes.

A covariance matrix should be a positive-definite matrix. For a covariance matrix to satisfy the positive-definite condition, its entire principal sub-matrix must have a positive determinant. The Gaussian distribution for an instance $\mathbf{x} = [\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_d]$ includes a determinant of covariance in its denominator as follows:

$$N(\mu, \Sigma) = \sqrt{((2\pi)^d \, |\Sigma|)}^{-1} \exp(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)). \tag{5.9}$$

If any determinant of the principal sub-covariance is not positive, some probability or marginal probability is infeasible or infinite. Therefore, this is contrary to the definition of probability, $0 \leq p \leq 1$. The Gaussian distribution should

therefore have a positive-definite covariance matrix. There are some causes and solutions when the covariance is not positive-definite. First, if some attributes in the dataset take on only a single value, then these attributes will have zero variance and the covariance will not be positive-definite (Wothke, 1993). These attributes do not contribute to the Bayesian classifier. Second, the absence of a positive-definite covariance may indicate perfect linear dependence of some attributes on another attribute. It may be easy to detect such relationships between two attributes on observation of a correlation coefficient matrix. However, the presence of multivariate dependencies, where several attributes together perfectly predict another attribute, may not be obvious. Third, when the sample size is small, the sample covariance may not be positive-definite due to mere sampling fluctuation (Anderson & Gerbing, 1984). In such cases, the offending covariance estimate could be fixed to zero with minimal harm to the Gaussian distribution (Gerbing & Anderson, 1987), and in such cases, it is equivalent to NB. Beyond these scenarios, missing data may cause the estimating technique on a pairwise basis to yield covariance that is not positive-definite. As with many problems, there are ways to sidestep this problem without actually trying to discern their cause. For example, spectral decomposition with an eigenvalue is generally used to make covariance positive-definite (Kupiec, 1998; Rebonato & Jackel, 2000; Wiesel et al., 2010).

5.3 Finding the best CGBN

To construct an efficient and accurate Bayesian network classifier, we employ a hybrid approach to the filter and wrapper methods on an undirected graphical model as shown in Table 5.1. We use the undirected graphical model because identifying explicit Bayesian factor among attributes, as in the case of Bayesian classifiers, is unnecessary if our goal is simply to compute the class-conditional probability of an instance in a BN. In order to benefit from both the rapidity of a filter type and the accuracy of a wrapper type, we use a hybrid approach that reduces the search space of BNs by applying a filter-type approach and finally selects the best BN in terms of classification accuracy. We propose a correlation-based Gaussian Bayesian network (CGBN), a kind of Gaussian Bayesian network that clusters attributes based on a weighted mean of an intra-class coefficient of determination.

Table 5.1: Strength of CGBN

Category	Description
Filter-based property	It generates a number of CGBNs that cluster the attributes based on the weighted mean of within-class coefficient of determination by using single linkage hierarchical clustering (SLHC)
Wrapper-based property	It searches for the best CGBN (bCGBN) having maximum classification accuracy among the generated CGBNs

We generate several CGBNs using single-linkage hierarchical clustering (SLHC), estimate a class-conditional Gaussian distribution function by considering the clustering result, and search for the best CGBN (bCGBN) with the

maximum classification accuracy. The class-conditional Gaussian distribution function is estimated by assuming class-conditional dependence within a single cluster and class-conditional independence among clusters. The procedure for finding bCGBN is shown in Figure 5.3.

Inputs: $\{(\mathbf{x}_i, y_i), i = 1, ..., n\}$, the dataset
Output: best CGBN
Step 1: Calculate R_c^2
For each combination (X_i, X_j) of Xs

$$r_c^2(\mathrm{X_i}, \mathrm{X_j}) = \sum_{k=1, y(\mathbf{x}) \in c_k}^{K} p(c_k) \times r^2(\mathrm{X_i}, \mathrm{X_j})$$

Step 2: Generate d CGBNs using SLHC with R_c^2
For each stage of SLHC
A single CGBN is generated from the updated set of attribute clusters

Step 3: Search for the bCGBN by using z-stages at equal intervals and Fibonacci search (z-SEI & FS)
Sub-step 1: Select primary CGBN using z-SEI search
Sub-step 2: Select the bCGBN using FS from the primary CGBN

Figure 5.3: Computation algorithm for finding bCGBN

5.3.1 Step 1: Calculate $\overline{R_c^2}$ to measure the dependence

We first calculate the intra-class coefficient of determination (r_c^2) for each pair combination of attributes (Xs), and use $\overline{r_c^2}$, the class frequency weighted mean of r_c^2 for all class values, as a measure of dependence between any two attributes. The coefficient of determination (r^2), or the square of the correlation coefficient,

is a measure of the dependence between two attributes, and is formulated as the square of the covariance divided by the product of their respective variances. It yields a value between 0 and 1, inclusive. Further, r^2 is strongly related to the covariance of the Gaussian distribution. Since the strength of the dependence can differ depending on the class value, we calculate r_c^2 separately for each class value. We calculate $\overline{r_c^2(\mathrm{X_i}, \mathrm{X_j})}$, the class frequency weighted mean of r_c^2, between X_i and X_j, for all class values, as in Eq. (5.10), for data with d attributes and K class values, and finally form a d $\times$ d weighted mean of an intra-class coefficient of determination matrix for all attributes ($\overline{R_c^2}$):

$$r_c^2(\mathrm{X_i}, \mathrm{X_j}) = \sum_{k=1, y(\mathbf{x}) \in c_k}^{K} p(c_k) \times r^2(\mathrm{X_i}, \mathrm{X_j}). \tag{5.10}$$

We calculated $\overline{r_c^2}$ and r^2 among attributes for the Iris dataset (Lichman, 2013). The Iris dataset has the distribution for each pair combination of attributes shown in Figure 5.4. It consists of three class values (Setosa, Versicolour, and Virginica) and four numerical attributes (sepal length (SL), pedal length (PL), sepal width (SW), and pedal width (PW)). In the Iris dataset, some pairs of attributes have different strengths of dependence between data of a single class value and those of all class values. Table 5.2 shows the distribution of attributes for each class value in the Iris dataset. SL and SW have high dependence within the same class value, but they have low dependence for all class values. They have a small value for r^2 (approximately 0.012) but a comparatively large value for $\overline{r_c^2}$ (approximately 0.348). At the same time, SL and PW have low dependence within the same class value but high dependence for all class values. They have a large value of r^2 (approximately 0.669) and

a comparatively value for small $\overline{r_c^2}$ (approximately 0.152). We can see that $\overline{r_c^2}$ differs from r^2, and effectively reflects dependence within the same class value rather than in r^2.

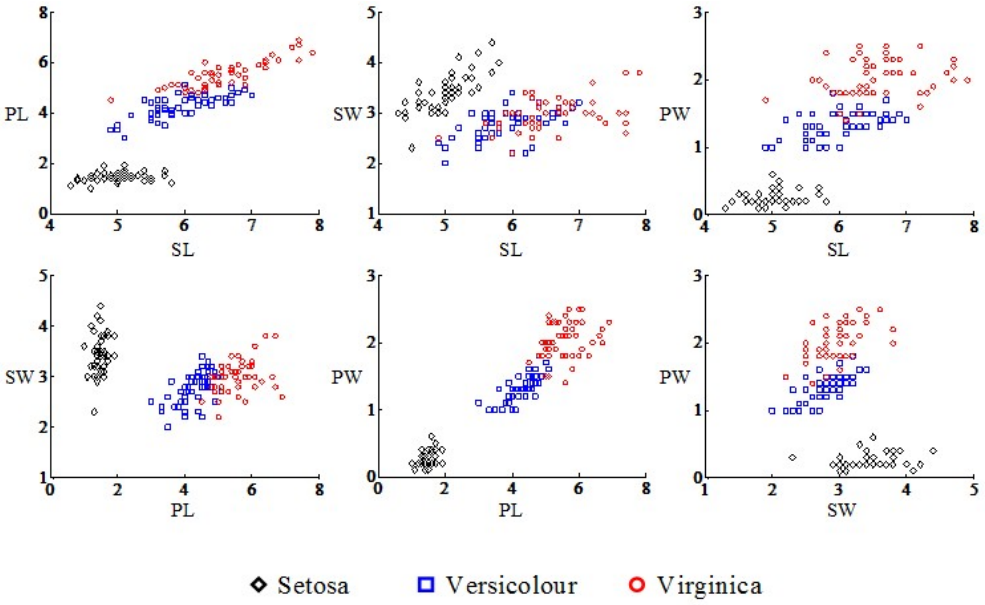


Figure 5.4: Distribution of attributes for each class value in the Iris dataset

Table 5.2: r_c^2 and r^2 of the Iris dataset

(a) r_c^2

	SL	PL	SW	PW
SL		0.462	0.348	0.152
PL			0.169	0.272
SW				0.269

(b) r^2

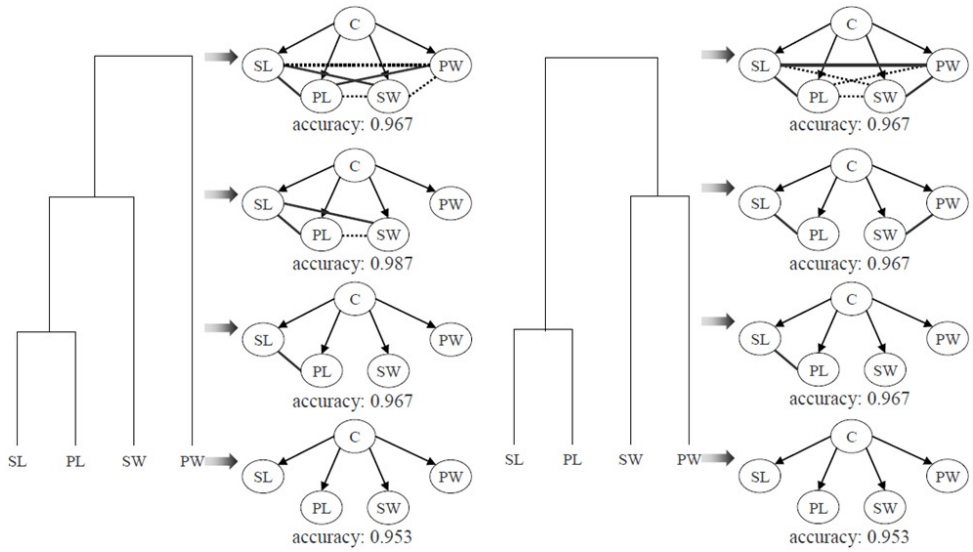
	SL	PL	SW	PW
SL		0.760	0.012	0.669
PL			0.177	0.927
SW				0.127

5.3.2 Step 2: Generate CGBNs using SLHC based on $\overline{R_c^2}$

We carried out single-linkage hierarchical clustering (SLHC) with $\overline{r_c^2}$ as a proximity measure to generate CGBNs and select the bCGBN with the maximum classification accuracy among them. A set of attribute clusters at each stage of the SLHC corresponds to a single CGBN. SLHC creates d stages, where d is the number of attributes, and iteratively generates a set of attribute clusters. SLHC starts with the structure of each attribute as a separate cluster equivalent to NB, merges the pair of clusters sharing the strongest dependence (maximum proximity) stage by stage, and forms a single cluster consisting of all attributes in the final stage, which is equivalent to a BQDA. We then assess the classification accuracy of each CGBN using a ten-fold cross-validation technique, and select the bCGBN with the maximum accuracy. When more than two CGBNs have the same value of maximum accuracy, we select the CGBN generated earlier.

The SLHC is a kind of graph-based clustering algorithm that defines proximity between two clusters as the proximity between the closest two points belong to different clusters. At each stage, SLHC merges two clusters whose attributes share the maximum value of $\overline{r_c^2}$ among attributes belonging to different clusters. Unlike SLHC, however, another HC algorithm, complete-linkage hierarchical clustering (CLHC), does not ensure such an outcome. Figure 5.5 shows the results of SLHC and CLHC on attributes with $\overline{r_c^2}$ for the Iris dataset. Both SLHC and CLHC involve four stages, and merge SL and PL sharing the maximum value of $\overline{r_c^2}$ in the second stage. However, SLHC and CLHC have different

clustering results in the third stage. In this stage, SLHC merges clusters with SL and SW that share the maximum value of $\overline{r_c^2}$ among attributes belonging to different clusters. By contrast, CLHC merges the SW and PW that maximise the smallest value of $\overline{r_c^2}$ between any two attributes belonging to different clusters. CGBN, with maximum accuracy, differs from SLHC and CLHC. In SLHC, the CGBN {{SL, PL, SW}, {PW}} generated in the third stage has a maximum classification accuracy of 0.987. However, in CLHC, the CGBN {{SL, PL}, {SW}, {PW}} generated in the second stage has a maximum classification accuracy of 0.967, which is lower than the maximum accuracy among CGBNs in SLHC.



(a) SLHC (b) CLHC

Figure 5.5: Generation of CGBNs using the Iris dataset

5.3.3 Step 3: Search for bCGBN using z-SEI and FS

The larger the number of attributes, the higher the computational complexity of calculating all CGBNs generated at all stages of the SLHC. In order to reduce computational complexity, we propose the use of z-stages at equal intervals and a Fibonacci search (z-SEI & FS) for the stages of the SLHG. This is a method to calculate CGBNs generated only in parts of the stages of the SLHC by using two search steps, as shown in Figure 5.6. If there are several attributes (if d> z), first, the z-SEI search is used to calculate only z CGBNs generated at stages at a constant interval I, which is calculated as (d−1)/(z−1). The search selects the "primary SLHC stage" generating CGBNs with the maximum accuracy among the z CGBNs. The SLHC search stages 1, 1+I, 1+2× I, . . . , 1+d−1. For example, when applying the z-SEI search (using z=7) to data with 21 attributes, I is (21−1)/(7−1), and the search order for the SLCH stages is 1, 4, 8, 11, 14, 18, and 21.

The FS (Ferguson, 1960; Overholt, 1973) finds the bCGBN with the maximum classification accuracy based on the primary stage of the SLHC. FS is relatively good at finding the value of the independent variable (SLHC stage order in this study) corresponding to the maximum of a unimodal function, and has an efficient computational complexity of $\log_2$ (N. independent variables). However, it may search the value locally for the maximum if the relation between the independent variable and the dependent variable is not a unimodal function. In FS, for the initial interval of values $[a_0, b_0]$, tolerance ε should be provided. FS continues to reduce the relevant interval from the initial search

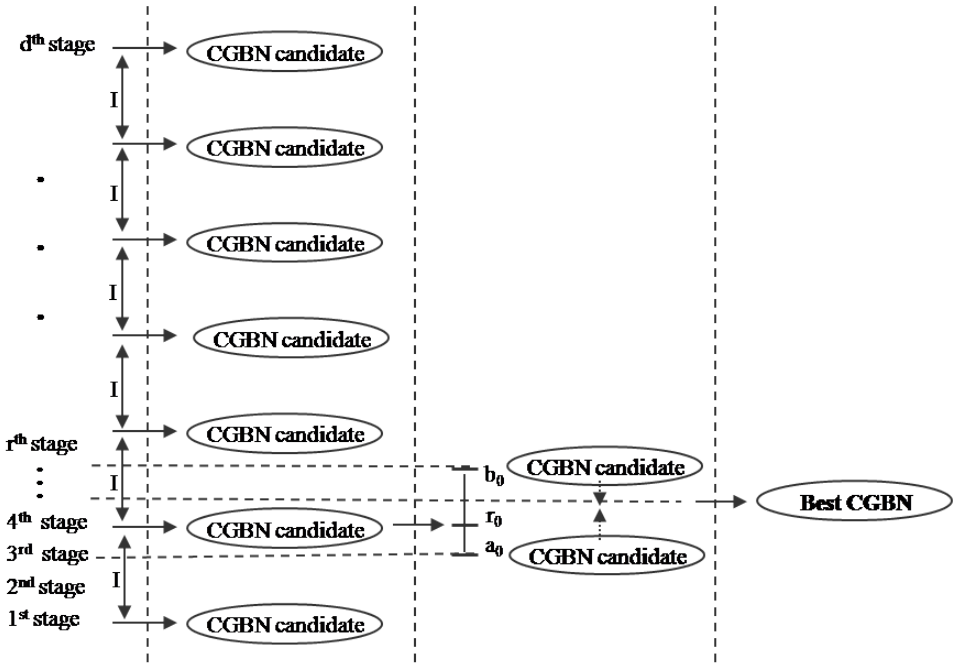


Figure 5.6: Searching for the SLHC stage corresponding to bCGBN: (a) z-SEI search, (b) Fibonacci search

$[a_0, b_0]$ (i.e., $b_0 - a_0 = I$), and is terminated when it reaches the ith search interval $|a_i - b_i| < \varepsilon$. We set ε to 1, and I to (d−1)/(r−1), which is used in the z-SEI search. We determine $a_0 and b_0$ from the primary stage of the SLHC (s_0) and I as follows:

$$a_0 = s_0 - (1 - r_o) \times I, \tag{5.11}$$

$$b_0 = s_0 + r_o \times I. \tag{5.12}$$

According to the FS algorithm, the search ratio (r_o) is calculated as F_{q-1} /F_q

after checking the smallest value of F_q (q^{th} Fibonacci number) satisfying

$$F_q > I/\varepsilon. \tag{5.13}$$

3.4. Estimation of class-conditional probability using attribute clusters

In CGBNs, class-conditional probability is estimated by assuming that attributes within a cluster are correlated and those belonging to different clusters are independent of one another. class-conditional probability in a CGBN is estimated as follows:

$$\begin{aligned} P(x_1, ..., x_d/c) &= \prod_{m=1}^{M} P(g_m/c) \\ &= \prod_{m=1}^{M} (\sqrt{2\pi})^{-n_m} (\sqrt{|\Sigma_{g_m/c}|})^{-1} \exp(-\frac{1}{2}(g_m - \mu_{g_m/c})' \Sigma_{g_m/c} (g_m - \mu_{g_m/c})) \end{aligned} \tag{5.14}$$

where $n_m$denotes the number of attributes belonging to the m^{th} cluster. Like FSSJ and BSEJ, a CGBN calculates class-conditional probability of all attributes by multiplying the class-conditional probability value for each attribute cluster (g_m) estimated through the application of the Gaussian distribution function. The parameters of the Gaussian distribution — the mean (μ_{g_m}) and the covariance (Σ_{g_m}) — are estimated through a maximum likelihood method by using training data. While a general BN of a DAG should estimate parent-conditional probabilities as the number of the attributes, the CGBN estimates the multivariate Gaussian probabilities as the number of the clusters.

If the covariance of attributes within a cluster is not positive definite in a CGBN, we remove some attributes, or transform the covariance to satisfy

the positive-definite condition. If an attribute takes only a single value, or is perfectly linearly dependent on another attribute, we remove it. Moreover, if an attribute takes only a single value of some class values, we substitute their variance values with a small value (0.00001). Further, if an attribute is perfectly linearly dependent on another attribute within some class value, we remove the attribute from those class values. If the covariance of clustered attributes is not positive definite having passed through the above processes, we employ spectral decomposition with eigenvalues (Rebonato & Jackel, 2000).

5.3.4 Applying BSE to bCGBN

In this study, we applied BSE to the bCGBN to improve classification accuracy. While the bCGBN treats the effect of redundant attributes by applying a multivariate Gaussian distribution to highly correlated attributes, it may unnecessarily include redundant attributes. Furthermore, the bCGBN can include irrelevant attributes for classification. Therefore, applying an attribute selection process, such as BSE to the bCGBN, can improve classification accuracy by removing attributes that do not contribute to the classification.

Figure 5.7 shows the process by which BSE is applied to the bCGBN. We start with the bCGBN with all the relevant attributes. We observe that removing one of the attributes in the bCGBN can improve accuracy relative to the bCGBN at the time, and so remove the attribute whose absence yields the maximum improvement in accuracy. The process is repeated for as long as improvement results from removals. If no removal results in an improvement, the algorithm is terminated, and the final updated bCGBN is returned. In the worst

case, the complexity of the algorithm is of the order $O(njd^2)$ in BSE where n is the number of training examples and j is the number of classes.

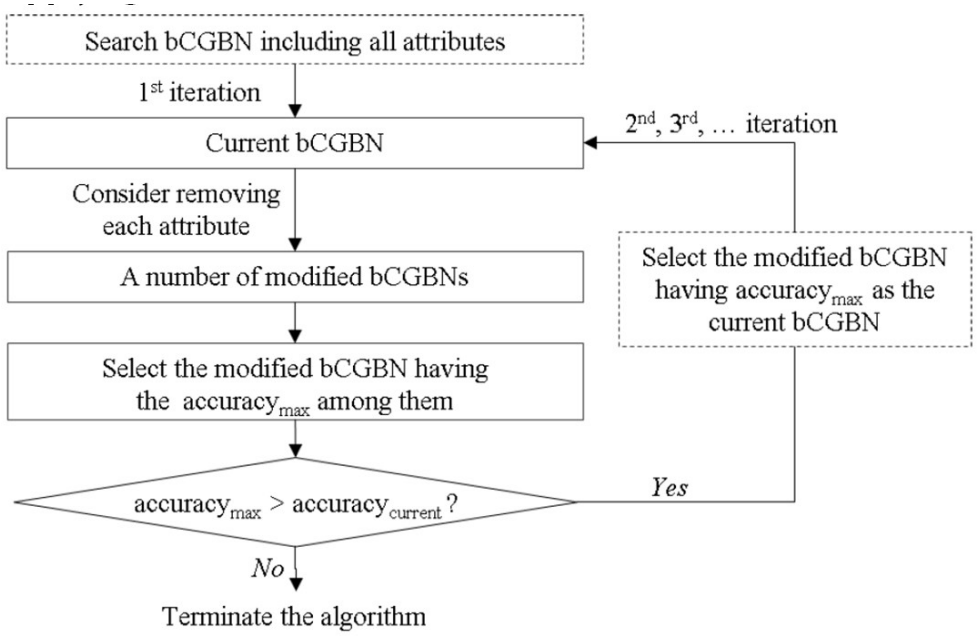


Figure 5.7: Process by which BSE is applied to bCGBN

In NB-BSE, BQDA-BSE, BSEJ, and bCGBN-BSE, we iteratively calculated various attribute sets for feature selection. Thus, the larger the computational complexity, the greater the number of attributes was. BSEJ has the largest, at $O(njd^3 + jv)$ where v is the number of combinations of attribute values, and the other algorithms using BSE are of the order $O(njd^2)$ (Zheng & Webb, 2005). All these classifiers measure the accuracy of attribute sets using 10-fold cross-validation, and select the attribute set with the maximum accuracy.

5.4 Performance on benchmark datasets

5.4.1 Experimental settings

We conducted experiments to verify the effectiveness of using the positive class prevalence estimate for calibration and classification tasks. Positive class prevalence for test set was estimated by SAC. The positive class prevalence estimate should be useful, provided that we have a training set where the positive class proportion is noticeably different from that in the test set. Therefore, the verification of our method required performance evaluation over a broad spectrum of test sets with different proportions of positive classes instead of a single test set. To vary the test conditions, we followed the experimental methodologies described in Chapter 2. We created test conditions where the proportion of the positive class was altered in a dataset in order to evaluate prevalence estimation methods. Test conditions included specific positive proportions ranging from 0% to 100%, in steps of 10%, and a positive proportion identical to that of the training set.

We conduct experiment with well-known Bayes classifiers, bCGBN and bCGBN-BSE. To estimate the performance of the classification, ten-fold cross validation was employed in our experiment. The Bayes classifiers used for comparison were divided into two categories. While NB, BQDA, TAN, and bCGBN were used for all of the attributes; NB-BSE, BQDA-BSE, BSEJ, and bCGBN-BSE employed attribute elimination. We set the z as 7 for z-SEI in bCGBN and bCGBN-BSE. The Gaussian distribution was applied to determine the class-conditional probability estimation for all algorithms. As preprocessing for all

datasets, we removed attributes that exhibited a constant value and perfectly linear dependence with another attribute. If the covariance within a single class value was not positive-definite, we employed Spectral Decomposition with Eigen Values (Rebonato & Jackel, 2000).

5.4.2 Experimental results

Table 5.3 summarizes the BS obtained for each calibration method over all the binary-class datasets. First, the average of BS was calculated from all datasets for each calibration method. Then, for each method, we counted the cases in which it had the best BS among all calibration methods across all datasets. Among all the compared calibration methods, the proposed RB method was the best across the datasets. The superiority of RB was decisive in terms of the abovementioned average and number of datasets on which its performance was the best, for every Bayesian classification algorithm. IR was second-best among all the classification algorithms. RB outperformed all the other methods on 61 73 out of the total of 87 datasets for every classification algorithm.

Table 5.4 summarizes the accuracy obtained when using different class prior assumptions. The one is to use the positive class prevalence (Prev.) of training set as class prior for the Bayesian classification models. The other is to use prevalence estimate for test set by SAC as class prior. To compare two methods, first, the average of accuracy was calculated from all datasets for each calibration method. Then, for each method, we counted the cases in which it had the best accuracy for each class prior assumption across all datasets. In all the Bayesian classification algorithms except for NB, the use of prevalence

estimate for test set outperformed the use of the prevalence of training set. But in NB, two methods showed equivalent results since the effect of class prior is much weaker than class conditional probability. It's because that the attributes correlated each other are reflected into a class conditional probability with duplication in NB. Difference in accuracy average ranged from 0.025 to 0.057 except for NB. And use of prevalence estimate for test set outperformed the other methods on 61 73 out of the total of 87 datasets for every classification algorithm excluding NB.

Table 5.3: Summary of BS of the calibration methods including ROC binning which use prevalence estimate

Classfier	Measure	HB	PS	IR	RB
NB	Average	0.151	0.209	0.145	0.098
	N. best datasets	4	2	8	73
QDA	Average	0.189	0.223	0.203	0.152
	N. best datasets	8	13	6	60
TAN	Average	0.185	0.220	0.187	0.146
	N. best datasets	7	7	7	66
bCGBN	Average	0.135	0.191	0.134	0.107
	N. best datasets	9	9	5	64
NB-BSE	Average	0.145	0.176	0.137	0.095
	N. best datasets	6	4	10	67
BQDA-BSE	Average	0.170	0.203	0.167	0.129
	N. best datasets	6	11	9	61
BSEJ	Average	0.138	0.195	0.129	0.092
	N. best datasets	3	4	10	70
bCGBN-BSE	Average	0.130	0.180	0.124	0.097
	N. best datasets	6	10	7	64

‘

Table 5.4: Classification accuracy and Wilcoxon signed-rank test result at α=0.05

Classfier	Measure	Class prior		Rank
		Training Prev.	Estimated Prev.	test
NB	Average	0.821	0.821	accepted
	N. best datasets	87	87	(not different)
QDA	Average	0.807	0.846	rejected
	N. best datasets	10	79	
TAN	Average	0.790	0.847	rejected
	N. best datasets	11	77	
bCGBN	Average	0.858	0.882	rejected
	N. best datasets	8	80	
NB-BSE	Average	0.827	0.865	rejected
	N. best datasets	13	76	
QDA-BSE	Average	0.834	0.870	rejected
	N. best datasets	12	77	
BSEJ	Average	0.848	0.874	rejected
	N. best datasets	16	72	
CGBN-BSE	Average	0.871	0.896	rejected
	N. best datasets	13	75	

5.5 Summary

Prevalence estimate result was applied as class prior for obtaining the calibrated probability and classifying instances using Bayes classifiers. In Bayesian classification algorithms, prior probability for a class value directly affects the classifi-

cation decision. The class prior is intrinsically identical to class prevalence. The results of experiments indicated that the use of the prevalence estimate generates calibrated probabilities and class predictions of a higher accuracy than other approaches that rely on the class prevalence of the training set, provided that we have a training set in which the positive proportion differs from that of the test set.

Additionally, we proposed bCGBN that considers both classification accuracy and intrinsic dependence between attributes. bCGBN has a BN structure of attribute clusters and is a type of semi-naïve Bayesian classifier that fixes a cluster structure from the undirected graphical model, similar to FSSJ and BSEJ. However, while FSSJ and BSEJ only use classification accuracy to make attribute clusters, in addition to considering classification accuracy, the proposed bCGBN considers intrinsic dependence between attributes with $\overline{r_c^2}$. We generated CGBNs by conducting SLHC for attributes and selected the bCGBN having the maximum classification accuracy among them. This study also employed bCGBN-BSE, a method that applies BSE to bCGBN. It required a complexity $O(njd^2)$ that is less than the complexity $O(njd^3 + jv)$ of FSSJ and BSEJ. The proposed bCGBN and bCGBN-BSE methods outperformed the previously introduced Bayes classifiers as indicated by the results of experiments on real-world problems.

Chapter 6

Conclusion

6.1 Contributions

In the class imbalance and class overlap problem, the direct application of the predictions of a classification model may lead to substantial error. This is because most classification algorithms are apt to focus on the classification of the majority instances while ignoring or misclassifying the minority instances to achieve high generalization capability. Apart from this, overlapping classes complicate the solution of class imbalance. In a class overlap problem, the instances sharing similar attribute values may belong to different classes. The combination of class imbalance and class overlap makes the classification problem more difficult and misleading. In situations where class imbalance and class overlap trigger an imperfect classification, thereby causing substantial bias in predicting classes, an analytical framework capable of estimating the minority proportion can serve as a more suitable tool by providing more reliable information to identify the real-world probability of being minority. The method for estimating minority proportions secure its feasibility when it is able to provide a good estimation of these proportions of the test set, provided that we have

Table 6.1: Methods and applications covered in this dissertation

Objective	Category	Topic	Substance
Obtaining calibrated probability	Method	ROC Binning (Chapter 2)	An accurate calibrated probability was obtained by using ROC framework
	Application	Predicting maladjusted soldiers (Chapter 3)	This study confirmed the effectiveness of ROC-based Binning for predicting maladjusted soldiers with MPI
Predicting prevalence	Method	SAC (Chapter 4)	Prevalence was predicted accurately by using training instances similar to test instances
	Application	Class prior of Bayesian classifier (Chapter 5)	This study extended the usability of the prevalence estimate as the parameter value of data mining algorithms

a training set in which the minority proportion is different from that of the test set. Table 6.1 summarized the methods and applications covered in this dissertation.

This study proposed a robust calibration technique, ROC Binning, to obtain the calibrated probability accurately even when the test and training sets have different a prevalence of positives. The technique uses TPR and FPR which are insensitive to class skews, and able to directly reflect the prevalence of positives in the test set. This method distinguishes the distribution nature within a class value and the effect of the class prevalence by using an ROC curve; thus, it performs robustly in situations with class skews. The effectiveness of ROC Binning was verified by comparing the technique to well-known calibration methods (Histogram Binning, Platt Scaling, and Isotonic Regres-

sion) using various classification algorithms as base-learner. We conducted experiments with five classification algorithms (NB, QDA, LDA, LR, and SVM) and 87 binary-class benchmark datasets in which the test positive prevalence was changed from 0% to 100%. Among the calibration methods, the proposed ROC Binning method was the outstanding front-runner across all the datasets in terms of Brier Score (BS) and Calibration Loss (CL). The proposed ROC Binning and its modification, namely TPR Binning, were applied to real-world MPI datasets. They were compared to the other calibration methods mentioned above. BS was also evaluated using the five classification algorithms as base-learner in the situation in which the prevalence of the test positive was changed. We were able to verify that all the classification algorithms resulted in similar characteristics. ROC Binning represented the best performance in terms of test prevalence. Moreover, there was little difference in performance between TPR Binning and ROC Binning for every classification algorithm. In summary, TPR Binning and ROC Binning outperformed previously proposed calibration methods for the MPI dataset in which the class imbalance and class overlap is severe and thus the classification performance is poor. Different management practices could be applied to different subject groups divided by the corresponding maladjusted proportion. For example, the subject group within the greater maladjusted proportion may require further psychiatric testing to ascertain their eligibility for exemption from military service. This study proposed a simple and effective prevalence estimation method, SAC, to estimate the prevalence accurately even if the test and training sets were to have a different class distribution. The proposed method is based on the AC method

and uses the TPR and FPR of training instances similar to the test instances. The proposed SAC adaptively uses a part of the training set for estimation of the TPR and FPR, whereas conventional AC requires the entire training set for the same purpose. The effectiveness of the SAC method was verified by comparing it to previously proposed prevalence estimation methods (AC, T50, X, MAX, and MS) using either SVM or PWK as base-learner. We conducted experiments with 87 binary-class benchmark datasets by changing the positive prevalence from 0% to 100%. Among the prevalence estimation methods, the proposed SAC was the outstanding front-runner across all the datasets in terms of Absolute Error (AE). Prevalence estimate result was applied as class prior for obtaining the calibrated probability and classifying instances using Bayes classifiers. In Bayesian classification algorithms, prior probability for a class value directly affects the classification decision. The class prior is intrinsically identical to class prevalence. The results of experiments indicated that the use of the prevalence estimate generates calibrated probabilities and class predictions of a higher accuracy than other approaches that rely on the class prevalence of the training set, provided that we have a training set in which the positive proportion differs from that of the test set. Additionally, we proposed a correlation-based Gaussian Bayesian network (CGBN), a hybrid Bayesian classification algorithm, which considers both classification accuracy and intrinsic dependence between attributes. CGBN has a BN structure for creating attribute clusters and is a type of semi-naïve Bayesian classifier that fixes a cluster structure from the undirected graphical model. The results of experiments indicated that good CGBNs are more accurate than the previously

proposed Bayes classifiers.

6.2 Future work

Here, we would like to address some limitations of this dissertation and indicate future research directions.In ROC Binning, non-overlapping bins were employed to obtain a discrete calibrated probability. For some data domains, however, overlapping bins would be more useful and effective. In the main task of obtaining the calibrated probability by the binning method, a dataset has to be split into several segments or bins (Bella et al., 2013). If too few bins are generated, the actual probabilities are not properly approximated to give detailed concrete information. However, if too many bins are defined, the actual probabilities are not properly estimated because of a wide fluctuation. A partial solution is to enable the bins to overlap.

And, instances were artificially divided into a predefined number of bins based on TPR plus FPR with Equal Width Interval Discretization in ROC Binning. However, some calibration methods, such as Isotonic Regression, can generate a number of segments by reflecting the natural distribution of classification function scores. The framework using ROC analysis, which obtains the calibrated probability using TPR and FPR distribution, is also freely applicable to other calibration method. The ROC framework is able to produce calibration methods to offer an accurate calibrated probability robust to positive prevalence change. In those cases, the generated instance subsets would have unequal intervals of TPR plus FPR unlike ROC Binning.

In this dissertation, we proposed ROC Binning and SAC, effective methods to estimate the minority proportion in terms of partition and aggregation respectively. They are able to support decision making on the basis of cost-minimization or benefit-maximization because they provide the actual probability or occurrence proportion for instances sharing similar characteristics instead of class prediction without a degree of confidence. These kinds of estimates can be used to make a cost-sensitive decision for each instance, when different costs are associated with different instances (Zadrozny & Elkan, 2001). Furthermore, to determine a reference value or cut-off point, we can conduct a Pareto analysis by calculating the cumulative percentages of the positive instances and total instances sorted by their classification function scores in descending order.

We assumed that class imbalance and class overlap lower the performance of conventional classification algorithms. However, other factors related to data complexity could also affect the classification performance. The universal data complexity is assessed as the Kolmogorov complexity of the mapping enforced by the dataset (L. Li, 2006) and a natural measure of the complexity of a problem is the error rate associated with a given classifier(Sotoca et al., 2005). However, it is also important to employ other measures that are less dependent on the chosen classifier itself but account for the intrinsic characteristics of a dataset. For example, the classification performance of a specific dataset would be strongly dependent on the other data complexity factor such as severability, geometry, and density.

We used a collection of datasets containing only numerical (real, integer) attributes to which to apply the proposed methods. However, data mining tech-

niques are increasingly required to deal with data that do not only consist of numerical attributes but also categorical attributes. Extending the proposed methods to process mixed-type datasets can improve their usability and widen the range of applications. Several methods have been proposed to treat datasets including categorical attributes (Agresti, 2013).

In this dissertation, the prevalence estimate result was used as class prior knowledge for obtaining the calibrated probability and classifying instances using Bayes classifiers. Incorporation of this prevalence knowledge when building the model could be the key element that allows an increase of performance in many applications. Prevalence can also be used to assign the cost of weight to individual instances in a training set to improve the model performance on the test set. Conventional class balanced learning in the training dataset does not carefully reflect a class skew of the test dataset, which can cause a significant estimation bias in real-world problems (Plessis & Sugiyama, 2014). If the prevalence, namely the class ratio of the test dataset, is estimated accurately, it is possible to effectively perform instance re-weighting or re-sampling to correct the systematic bias.

The MPI dataset was used to obtain the calibrated probability using ROC Binning and TPR Binning. There is a need to improve the reliability of the MPI test and identification of factors affecting maladjustment, because the current MPI test does not provide a perfect indication as to an individual's adaptability to military service. Further, the development of analytical algorithms more appropriate to the MPI can improve the performance of the model. Therefore, additional studies on the MPI are required for more effective management of

persons subject to conscription.

In this dissertation, the proposed methods were applied to a limited domain of the MPI and various benchmark UCI datasets. To make this research more fruitful, more real-world application domains would have to be explored and experimented upon. For example, estimating the fault rate with ROC Binning or SAC is very useful in predicting the quality management of product lines. Besides, medical diagnosis, text categorization, and risk/disease prediction require an estimation of the actual probability of being minority because it supports decision making by providing a reliable assessment of cost and effect. If we have a number of alternatives in decision making, we would be able to determine relative and absolute priorities in terms of cost-effectiveness using the calibrated probability and prevalence. According to the task description in this dissertation, ROC Binning and SAC can be successfully used to estimate the actual probability of being a minority.

Bibliography

Agoritsas, T., Courvoisier, D. S., Combescure, C., Deom, M., & Perneger, T. V. (2011). Post-test probability according to prevalence. *Journal of general internal medicine*, *26*(10), 1091-1091.

Agresti, A. (1996). *An introduction to categorical data analysis*. New York: John Wiley & Sons.

Agresti, A. (2013). *Categorical data analysis*. John Wiley & Sons.

Alejo, R., Valdovinos, R., García, V., & Pacheco-Sanchez, J. (2013). A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios. *Pattern Recognition Letters*, *34*(4), 380-388.

Amoroso, L. (1938). Vilfredo pareto. *Econometrica*, *6*(1), 21.

Anderson, J., & Gerbing, D. (1984). The effect of sampling error on convergence, improper solutions, and goodness of fit indices for maximum likelihood confirmatory factor analysis. *Psychometrika*, *49*(2), 155-173.

Auvray, V. (2007). *Contributions to Bayesian network learning (Dissertation)*. Université de Liège.

Baccianella, S., Esuli, A., & Sebastiani, F. (2013). Variable-constraint classifi-

cation and quantification of radiology reports under the acr index. *Expert Systems with Applications*, *40*, 3441-3449.

Bandos, A. I., Rockette, H. E., & Gur, D. (2007). Exact bootstrap variances of the area under roc curve. *Communications in Statistics-Theory and Methods*, *36*(13), 2443-2461.

Barandela, R., Sánchez, J., García, V., & Rangel, E. (2003). Strategies for learning in class imbalance problems. *PatternRecognition*, *36*, 849-851.

Barranquero, J., González, P., Díez, J., & Coz, J. J. (2013). On the study of nearest neighbor algorithms for prevalence estimation in binary problems. *Pattern Recognition*, *46*, 472-482.

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, *6*(1), 20-29.

Bella, A., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M. J. (2013). On the effect of calibration in classifier combination. *Applied Intelligence*, *38*(4), 566-585.

Ben-Gal, I. (2007). Bayesian networks. In F. Ruggeri, R. Kenet, & F. Faltin (Eds.), *Encyclopedia of statistics in quality and reliability.* New York, NY, USA: John Wiley & Sons.

Brier, G. (1950). Verification of forecasts expressed in terms of probabilities. *Monthly Weather Review*, *78*, 1-3.

Buntine, W. (1992). Learning classification trees. *Statistics & Computing*, *2*(2), 63-73.

Burez, J., & Poel, D. V. d. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, *36*, 4626-4636.

Bøttcher, S. (2004). *Learning Bayesian networks with mixed variables (Dissertation).* Aalborg University.

Çalış, N., & Erol, H. (2012). A new per-field classification method using mixture discriminant analysis. *Journal of Applied Statistics*, *39*(10), 2129-2140.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 1-27.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321-357.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explorations*, *6*(1), 1-6.

Chen, C., Cheng, K., & Liao, H. (2004). Fairing of polygon meshes via Bayesian discriminant analysis. *Journal of WSCG*, *12*, 175-182.

Chickering, D., Geiger, D., & Heckerman, D. (1994). *Learning Bayesian networks is np-hard (technical report msr-tr-94-17).* Redmond, WA, USA:

Microsoft Research.

Choi, K. H., Jung, S. K., Choi, K. P., Moon, C. B., Kim, J. M., Park, B. K., ... Bae, J. K. (2009). *Development of new personality tests for military.* Seoul, ROK: Institute for Defense Analyses.

Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, *14*, 462-467.

Cohen, I., & Goldszmidt, M. (2004). *Properties and benefits of calibrated classifiers.*

Cooley, P. (1975). Bayesian and cost considerations for optimal classification with discriminant analysis. *The Journal of Risk and Insurance*, *42*(2), 277-287.

Cooper, G. (1990). Computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, *42*(2-3), 393-405.

Cortes, C., & Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273-297.

Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in Bayesian belief networks is np-hard. *Artificial Intelligence*, *60*(1), 141-153.

Das, B., Krishnan, N. C., & Cook, D. J. (2014). Handling imbalanced and over-

lapping classes in smart environments prompting dataset. *Data Mining for Service, Studies in Big Data*, *3*, 199-219.

Dash, R., Paramguru, R. L., & Dash, R. (2011). Comparative analysis of supervised and unsupervised discretization techniques. *International Journal of Advances in Science and Technology*, *2*(3), 29-37.

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, *837-845*.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1-130.

Denil, M., & Trappenberg, T. (2010). Overlap versus imbalance. In *Advances in artificial intelligence* (p. 220-231). Springer.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and science analysis.* New York: John Wiley & Sons.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification (2nd edition).* New York, NY, USA: John Wiley & Sons.

Duman, E., Ekinci, Y., & Tanrıverdi, A. (2012). Comparing alternative classifiers for database marketing: The case of imbalanced datasets. *Expert Systems with Applications*, *39*, 48-53.

Egan, J. P. (1975). *Signal detection theory and roc analysis, series in cognition*

and perception. New York: Academic Press.

Esuli, A., & Sebastiani, F. (2010). Sentiment quantification. *IEEE intelligent systems*, *25*(4), 72-79.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, *27*, 861-874.

Fawcett, T., & Flach, P. (2005). A response to webb and ting's on the application of roc analysis to predict classification performance under varying class distributions. *Machine Learning*, *58*, 33-38.

Fawcett, T., & Niculescu-Mizil, A. (2007). Pav and the roc convex hull. *Machine Learning*, *68*(1), 97-106.

Ferguson, D. E. (1960). Fibonaccian searching. *Communications of the ACM*, *3*(12), 648.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*(1), 27-38.

Fisher, R. (1936). The use of multiple measurements in taxonomy problems. *Annals of Eugenics*, *7*(2), 179-188.

Flach, P., & Matsubara, E. T. (2007). A simple lexicographic ranker and probability estimator. *Machine Learning: ECML 2007 Lecture Notes in Computer Science*, *4701*, 575-582.

Forman, G. (2008). Quantifying counts and costs via classification. *Data Mining Knowledge Discovery*, *17*, 164-206.

Forman, G., Kirshenbaum, E., & Suermondt, J. (2006). Pragmatic text mining: minimizing human effort to quantify many issues in call logs. In *Proceedings of the 12th acm SIGKDD international conference on knowledge discovery and data mining* (p. 852-861).

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, *11*(1), 86-92.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*, 131-163.

Friedman, N., & Koller, D. (2003). Being Bayesian about network structure, a Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, *50*, 95-125.

Galar, M., Fernandez, A., Barrenechea, E., & Bustince, H. (2012). A review on ensembles for the class imbalance problem: Bagging,boosting and hybrid-based approaches. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *42*(4), 463 - 484.

García, V., Alejo, R., Sánchez, J., Sotoca, J., & Mollineda, R. (2006). Combined effects of class imbalance and class overlap on instance-based classification. *Intelligent Data Engineering and Automated Learning – IDEAL*

2006 (Lecture Notes in Computer Science), *4224*, 371-378.

García, V., Mollineda, R. A., & Sánchez, J. S. (2008). On the k-nn performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis and Applications*, *11*, 269-280.

Garey, M., & Johnson, D. (1979). *Computers and intractability: A guide to the theory of np-completeness*. New York, NY, USA: W.H. Freeman & Co.

Gebel, M. (2009). *Multivariate calibration of classifier scores into the probability space (Dissertation)*. Technical University of Dortmund.

Geiger, D., & Heckerman, D. (1994). Learning gaussian networks. In *Proceedings of the 10th conference on uncertainty in artificial intelligence* (p. 235-243). Morgan Kaufman.

Gerbing, D., & Anderson, J. (1987). Improper solutions in the analysis of covariance structures: their interpretability and a comparison of alternate respecifications. *Psychometrika*, *52*(1), 99-111.

Gessel, I. M. (1996). Counting acyclic digraphs by sources and sinks. *Discrete Math*, *160*, 253-258.

Giudici, P., & Green, P. J. (1999). Decomposable graphical gaussian model determination. *Biometrika*, *86*(4), 785-801.

Goldberg, L. R. (1965). Diagnosticians vs. diagnostic signs: The diagnosis of psychosis vs. neurosis from the mmpi. *Psychological Monographs: General*

and Applied, *79*(9), 1-28.

Grossman, D., & Domingos, P. (2004). Learning Bayesian network classifiers by maximizing conditional likelihood. In *Proceeding of the 21st international conference on machine learning.*

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157-1182.

Gómez-Villegas, M. A., Maín, P., & Susi, R. (2007). Sensitivity analysis in gaussian Bayesian networks using a divergence measure. *Communications in Statistics-Theory and Methods*, *36*, 523-539.

Hambrush, S., & Tu, H. (1997). Edge weight reduction problems in directed acyclic graphs. *Journal of Algorithms*, *24*, 66-93.

Heckerman, D. (1995). *A tutorial on learning with Bayesian networks (technical report msr-tr-95-06).* Redmond, WA, USA: Microsoft Research.

Heckerman, D., Geiger, D., & Chickering, D. (1995). Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, *20*, 197-243.

Hermstein, C., & Murray, R. (1994). *The bell curve: intelligence and class structure in american life.* New York, NY, USA: The Free Press.

Huang, K., King, I., & Lyu, M. (2003). Finite mixture model of bounded semi-naïve Bayesian networks classifier. In *Proceedings of the 10th interna-*

tional conference on neural information processing (p. 115-122). Springer-Verlag.

Isonotic regression software (2005). Bern, Switzerland: University Bern.

Jakulin, A., & Irish, I. (2006). *Bayesian learning of markov network structure.* Berlin, GER: Springer.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429-449.

Jayech, K., & Mahjoub, M. (2011). Clustering and Bayesian network for image of faces classification. *International Journal of Advanced Computer Science and Applications*, *Special Issue on Image Processing and Analysis*, 35-44.

Johnson, R., & Wichern, D. (2002). *Applied multivariate statistical analysis.* Englewood Cliffs, NJ, USA: Prentice-Hall.

Kang, P., & Cho, S. (2008). Locally linear reconstruction for instance-based learning. *Pattern Recognition*, *41*(11), 3507-3518.

Kohavi, R., & Sommerfield, D. (1995). *Feature subset selection using the wrapper method: Overfitting and dynamic search space topology.* Menlo Park, CA, USA: Morgan Kaufmann.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Sci-*

ence and Engineering, *30*, 25-56.

Kupiec, P. (1998). Stress testing in a value at risk framework. *Journal of Derivatives*, *6*, 7-24.

Lambrou, A., Papadopoulos, H., Nouretdinov, I., & Gammerman, A. (2012). Reliable probability estimates based on support vector machines for large multiclass datasets. *Artificial Intelligence Applications and Innovations*, *382*, 182-191.

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. In *Proceedings of the 10th conference on uncertainty in artificial intelligence* (p. 399-406).

Lauritzen, S., & Spiegelhalter, D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, *50*(2), 157-224.

Lauritzen, S. L., & Wermuth, N. (1989). Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, *17*, 31-57.

Lee, H. J. (2007). *Novelty detection for class imbalance: LVQ-based algorithm and its application to security and crm (Dissertation).* Seoul National University.

Li, L. (2006). *Data complexity in machine learning and novel classification algorithms (Dissertation).* California Institute of Technology.

Li, S. (1995). *Markov random field modeling in computer vision.* London, UK: Springer-Verlag.

Liang, P., & Srebro, N. (2004). *Methods and experiments with bounded tree-width markov networks (technical report mit-csail-tr-2004-081)* Redmond, WA, USA: Microsoft Research.

Lichman, M. (2013). UCI machine learning repository [http://archive.ics.uci.edu/ml]. University of California, Irvine, School of Information and Computer Sciences.

Lin, H.-T., Lin, C.-J., & Weng, R. C. (2007). A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, *68*(3), 267-276.

Liskovets, V. (1975). On the number of maximal vertices of a random acyclic digraph. *Theory of Probability and its Applications*, *20*(2), 401-409.

Liu, X. Y., Wu, J., & Zhou, Z. H. (2009). Exploratory undersampling for class-imbalance learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *39*(2), 539 - 550.

López, V., Fernández, A., Moreno-Torres, J. G., & Herrera, F. (2012). Analysis of preprocessing vs. cost-sensitive learning for imbalanced classifcation. open problems on intrinsic data characteristics. *Expert Systems with Applications*, *39*, 6585-6608.

Martínez-Camblor, P., Carleos, C., & Corral., N. (2011). Powerful nonparametric statistics to compare k independent roc curves. *Journal of applied*

statistics, *38*(7), 1317-1332.

Matlab version 7.10.0. (2010). Natick, Massachusetts: The MathWorks Inc.

Mclachlan, G. (2004). *Discriminant analysis and statistical pattern recognition.* New York, NY, USA: John Wiley & Sons.

Medcalc version 14.8.1. (2014). Ostend, Belgium: MedCalc Software Bvba.

Mitchell, T. (1997). *Machine learning.* New York, NY, USA: McGraw Hill.

Murphy, A. H. (1972). Scalar and vector partitions of the probability score: Part ii. n-state situation. *Journal of Applied Meteorology*, *11*, 182-1192.

Murphy, A. H. (1973). A new vector partition of the probability score. *Journal of Applied Meteorology*, *12*(4), 595-600.

Naeini, M. P., Cooper, G. F., & Hauskrecht, M. (2014). Binary classifier calibration: Non-parametric approach. *arXiv preprint arXiv:1401.3390*.

Neapolitan, R. (2003). *Learning Bayesian networks* NJ, USA: Englewood Cliffs.

Nemenyi, P. (1963). *Distribution-free multiple comparisons (Dissertation).* Princeton University.

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on machine learning* (p. 625-632). ACM.

Overholt, K. J. (1973). Efficiency of the fibonacci search method. *Bit*, *13*, 92-96.

Pazzani, M. (1997). Searching for dependencies in Bayesian classifiers. In D. F. V & H. Lenz (Eds.), *Learning from data: artificial intelligence and statistics* (p. 239-248). New York, NY, USA: Springer-Verlag.

Pearl, J. (1988). *Probabilistic reasoning in intelligence systems: Networks of plausible inference*. Los Altos, CA, USA: Morgan Kaufmann.

Pepe, M. S. (2003). *The statistical evaluation of medical tests for classification and prediction*. New York: Oxford University Press.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *In: Advances in Large Margin Classifiers*, 61-74.

Plessis, M. C. D., & Sugiyama, M. (2014). Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, *50*, 110-119.

Prati, R. C., Batista, G. E. A. P. A., & Monard, M. C. (2004). Class imbalances versus class overlapping: an analysis of a learning system behavior. *MICAI 2004: Advances in Artificial Intelligence, Lecture Notes in Computer Science*, *2972*, 312-321.

Prim, R. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal*, *36*, 1389-1401.

Provost, F. J., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. *International Conference on Machine Learning*, *98*, 445-453.

Pérez, A., Larranaga, P., & Inza, I. (2006). Supervised classification with conditional gaussian networks: Increasing the structure complexity from naïve Bayes. *International Journal of Approximate Reasoning*, *43*, 1-25.

Rebonato, R., & Jackel, P. (2000). The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. *Journal of Risk*, *2*, 17-27.

Ren, H., Zhou, X.-H., & Liang, H. (2004). A flexible method for estimating the roc curve. *Journal of Applied Statistics*, *31*(7), 773-784.

Robinson, R. (1973). Counting labeled acyclic digraphs. In F. Harary (Ed.), *New directions in the theory of graphs* (p. 239-273). New York, NY, USA: Academic Press.

Sahami, M. (1996). limited dependence Bayesian classifiers. In *Proceedings of the 2nd international conference on knowledge discovery and data mining* (p. 335-338). AAAI press.

Scutari, M., & Brogini, A. (2012). Bayesian network structure learning with permutation tests. *Communications in Statistics-Theory and Methods*, *41*(16-17), 3233-3243.

Shimony, S. (1994). Finding maps for belief networks is np-hard. *Artificial*

Intelligence, *68*(2), 399-410.

Shin, H. J., & Cho, S. (2006). Response modeling with support vector machines. *Expert Systems with Applications*, *30*(4), 746-760.

Sotoca, J. M., Sanchez, J. S., & Mollineda, R. A. (2005). A review of data complexity measures and their applicability to pattern classification problems. In *Actas del iii taller nacional de mineria de datos y aprendizaje.–tamida* (p. 77-83).

Stanley, R. (1973). Acyclic orientations of graphs. *Discrete Math*, *5*(2), 171-178.

Sun, M., Choi, K., & Cho, S. (2015). Estimating the minority class proportion with the roc curve using military personality inventory data of the rok armed forces. *Journal of Applied Statistics*, *42*(8), 1677-1689.

Sun, Y., Kamel, M. S., Wong, A. K. C., & Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, *40*(12), 3358-3378.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science*, *240*, 1285-1293.

Swets, J. A., Dawes, R. M., & Monahan, J. (2000). Better decisions through science. *Scientific American*, *283*, 82-87.

Switzer, P. (1980). Extensions of linear discriminant analysis for statistical

classification of remotely sensed satellite imagery. *Journal of the International Association for Mathematical Geology*, *12*(4), 367-376.

Sánchez, J. S., Mollineda, R. A., & Sotoca, J. M. (2007). An analysis of how training data complexity affects the nearest neighbor classifiers. *Pattern Analysis & Applications*, *10*(3), 189-201.

Talavera, L. (2005). An evaluation of filter and wrapper methods for feature selection in categorical clustering. In *Proceedings of the 6th international symposium on intelligent data analysis* (p. 440-451).

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining* New York: Addison Wesley.

Tan, S. (2005). Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, *28*, 667-671.

Teyssier, M., & Koller, D. (2005). Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Proceedings of the 21st conference on uncertainty in artificial intelligence* (p. 584-591).

Wallace, B. C., & Dahabreh, I. J. (2012). Class probability estimates are unreliable for imbalanced data (and how to fix them). In *Ieee 12th international conference on data mining* (p. 695-704). IEEE Computer Society.

Webb, G., & Ting, K. (2005). On the application of roc analysis to predict classification performance under varying class distributions. *Machine Learning*, *58*, 25-32.

Weiss, G. M. (2004). Mining with rarity: A unifying framework. *SIGKDD Explorations*, *6*(1), 7-19.

Wiesel, A., Eldar, Y., & Hero, A. I. (2010). Covariance estimation in decomposable gaussian graphical models. *IEEE Trans. on Signal Processing*, *58*(3), 1482-1492.

Wothke, W. (1993). Non positive definite matrices in structural modeling. In K. Bollen & J. Long (Eds.), *Testing structural equation models* (p. 256-293). Newbury Park, CA, USA: Sage Publication.

Xiao, J., Xie, L., He, C., & Jiang, X. (2012). Dynamic classifier ensemble model for customer classification with imbalanced class distribution. *Expert Systems with Applications*, *39*, 3668-3675.

Xiong, H., Zhang, X., Zhang, Y., Ma, F., Li, Y., & Li, L. (2005). An investigation of the prevalence of depressive symptoms in soldiers during military training. *Prev Med*, *41*, 642-645.

Yang, H., & Carlin, D. (2000). Roc surface: a generalization of roc curve analysis. *Journal of Biopharmaceutical Statistics*, *10*(2), 183-196.

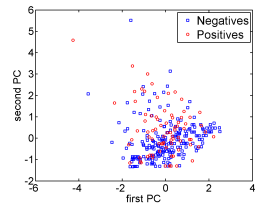
Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naïve Bayesian classiers. In *In proceedings of the eighteenth international conference on machine learning* (p. 609-616).

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *In proceedings of the eighth acm*

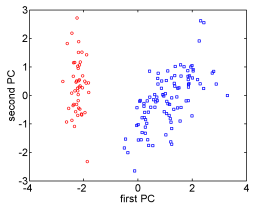
SIGKDD international conference on knowledge discovery and data mining (p. 694-699).

Zheng, F., & Webb, G. I. (2005). A comparative study of semi-naive bayes methods in classification learning. In *Proceedings of the fourth australasian data mining conference* (p. 141-156).

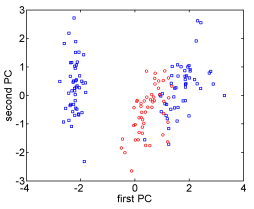
Appendix A: First and second PCs of the benchmark datasets



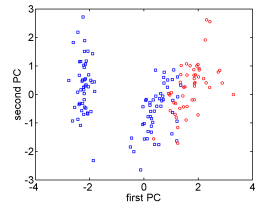
Haberman



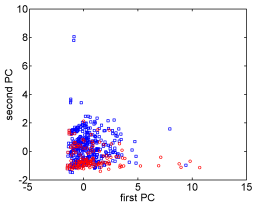
Iris.1



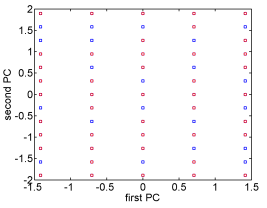
Iris.2



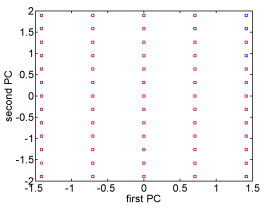
Iris.3



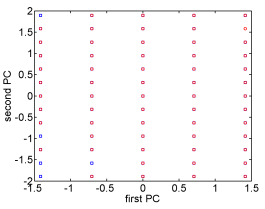
Blood



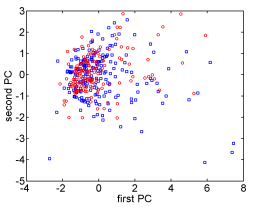
Balance.1



Balance.2

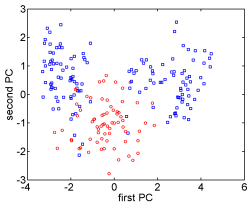


Balance.3

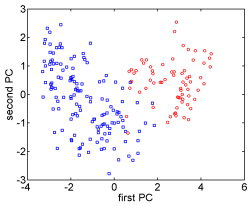


Liver

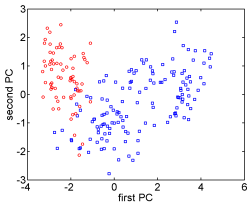
PCs per binary-class dataset



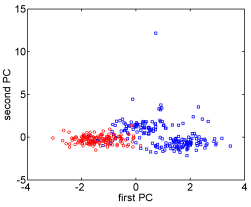
Seeds.1



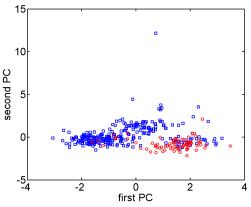
Seeds.2



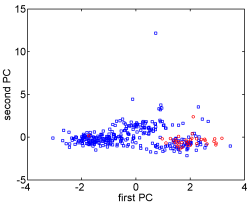
Seeds.3



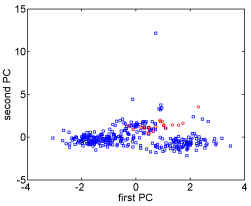
Ecoli.1



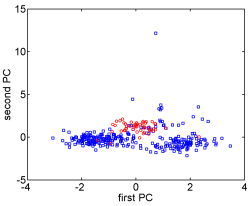
Ecoli.2



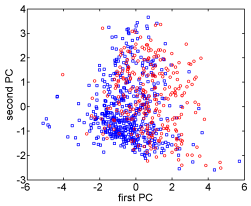
Ecoli.3



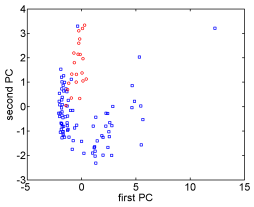
Ecoli.4



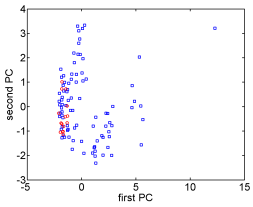
Ecoli.5



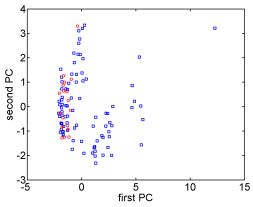
Pima



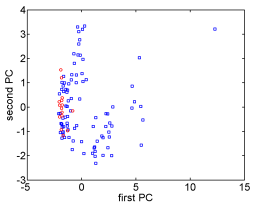
Breast.1



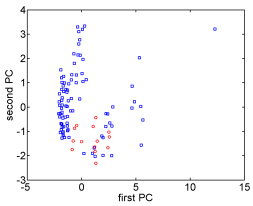
Breast.2



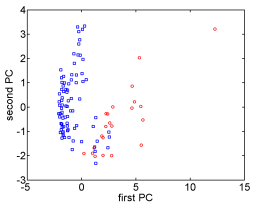
Breast.3



Breast.4

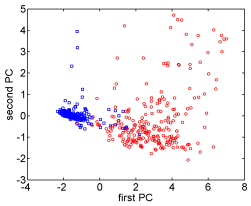


Breast.5

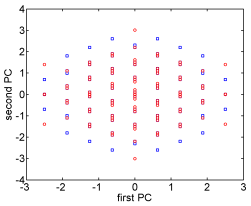


Breast.6

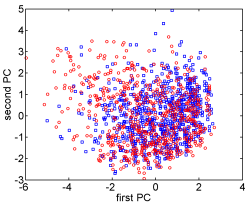
PCs per binary-class dataset



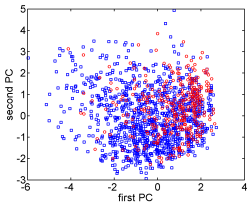
Wisconsin



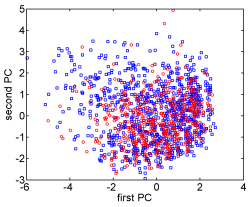
Tic-Tac-Toe



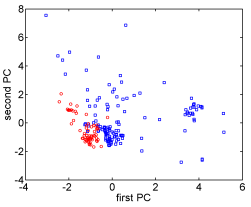
Contraceptive.1



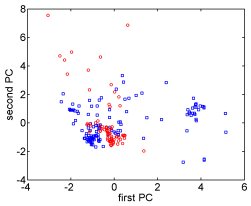
Contraceptive.2



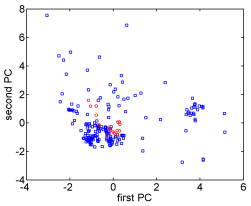
Contraceptive.3



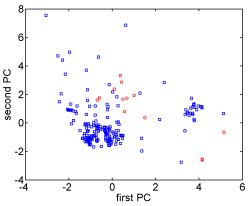
Glass.1



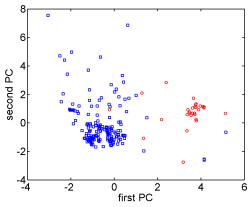
Glass.2



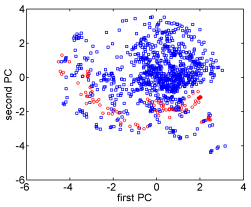
Glass.3



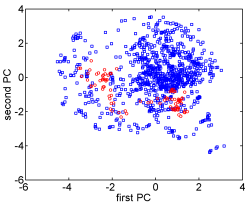
Glass.4



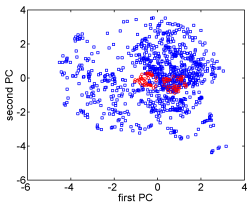
Glass.5



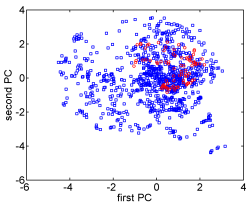
Vowel.1



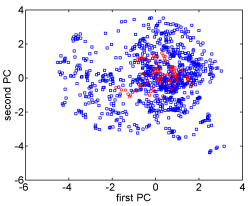
Vowel.2



Vowel.3

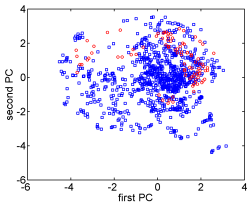


Vowel.4

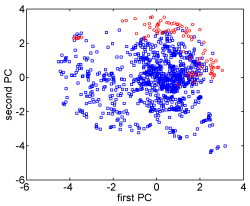


Vowel.5

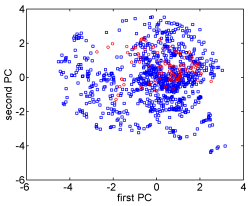
PCs per binary-class dataset



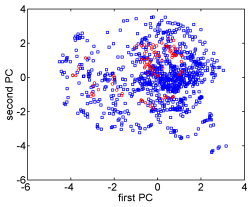
Vowel.6



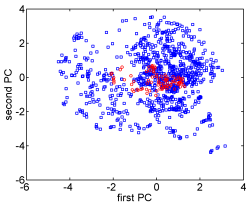
Vowel.7



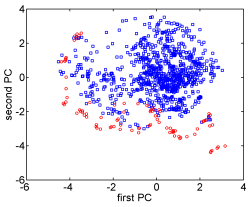
Vowel.8



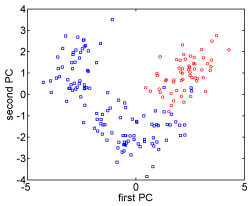
Vowel.9



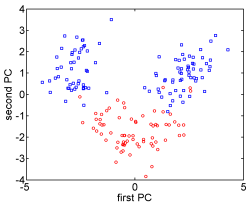
Vowel.10



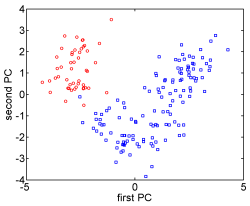
Vowel.11



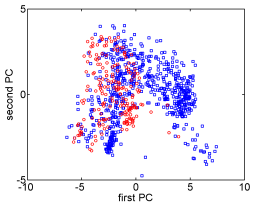
Wine.1



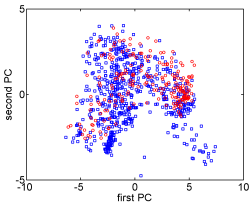
Wine.2



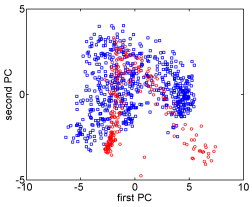
Wine.3



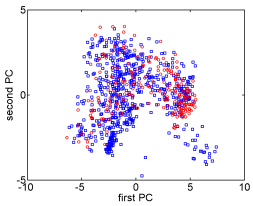
Statlog.1



Statlog.2



Statlog.3



Statlog.4

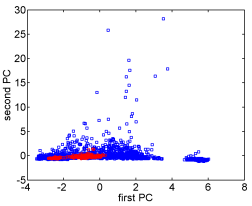


Image.1

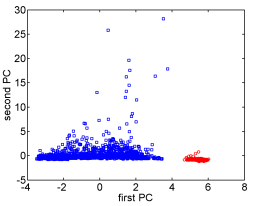


Image.2

PCs per binary-class dataset

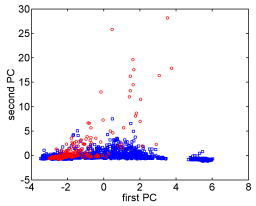


Image.3

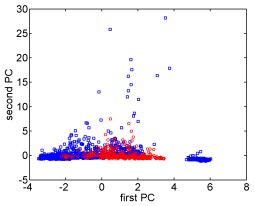


Image.4

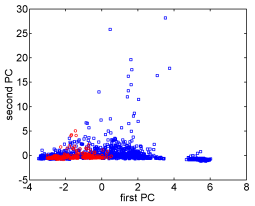


Image.5

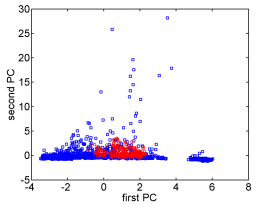


image.6

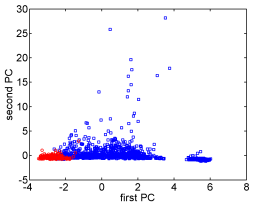
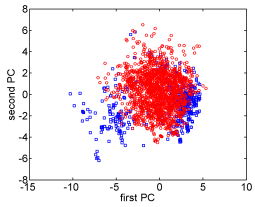
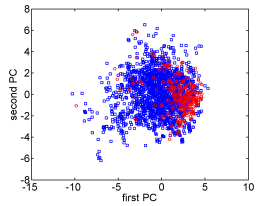


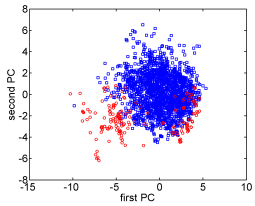
image.7



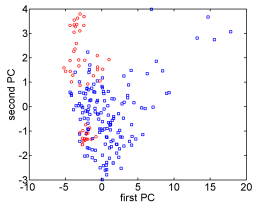
Cardio.1



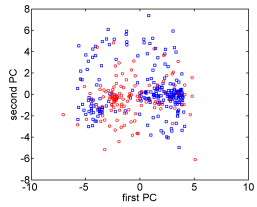
Cardio.2



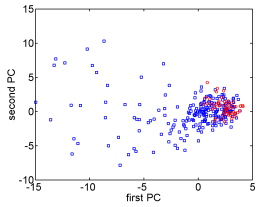
Cardio.3



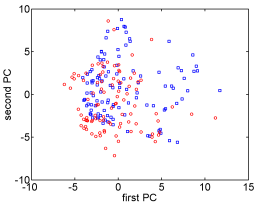
Parkinsons



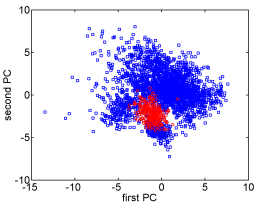
Ionosphere



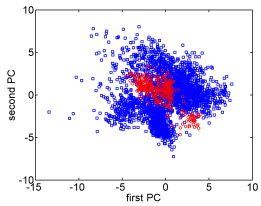
SPECTF



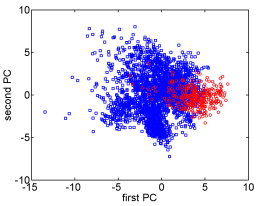
Sonar



Digits.1

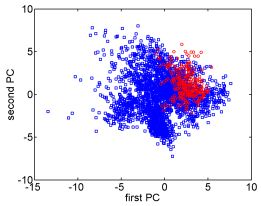


Digits.2

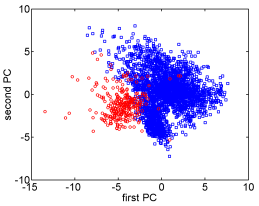


Digits.3

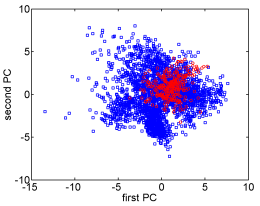
PCs per binary-class dataset



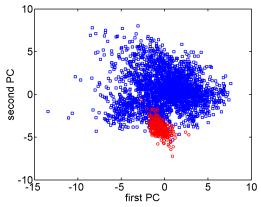
Digits.4



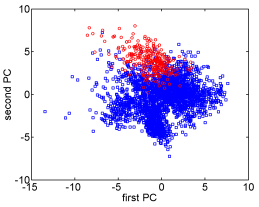
Digits.5



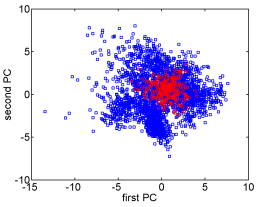
Digits.6



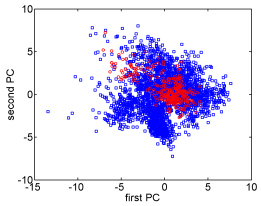
Digits.7



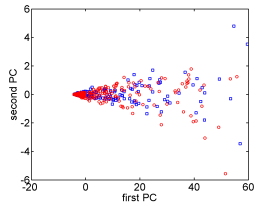
Digits.8



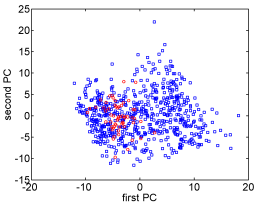
Digits.9



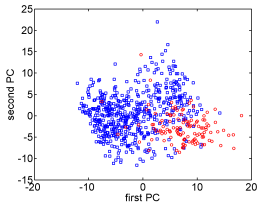
Digits.10



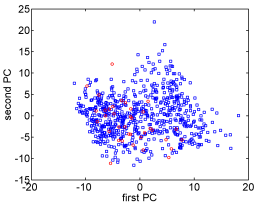
Hill-Valley



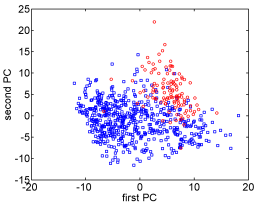
Urban.1



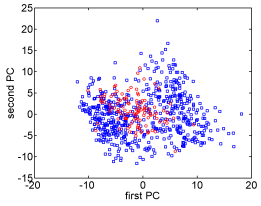
Urban.2



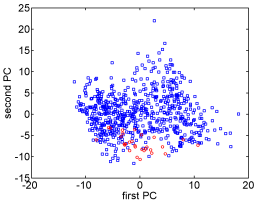
Urban.3



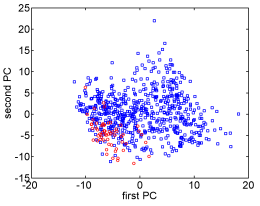
Urban.4



Urban.5

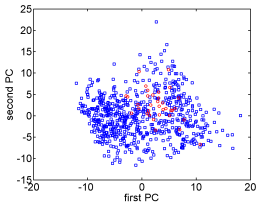


Urban.6

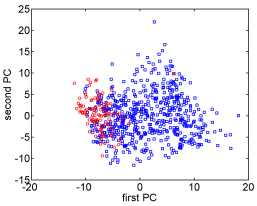


Urban.7

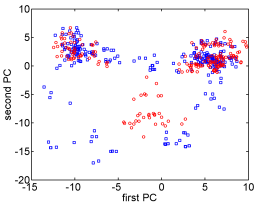
PCs per binary-class dataset



Urban.8



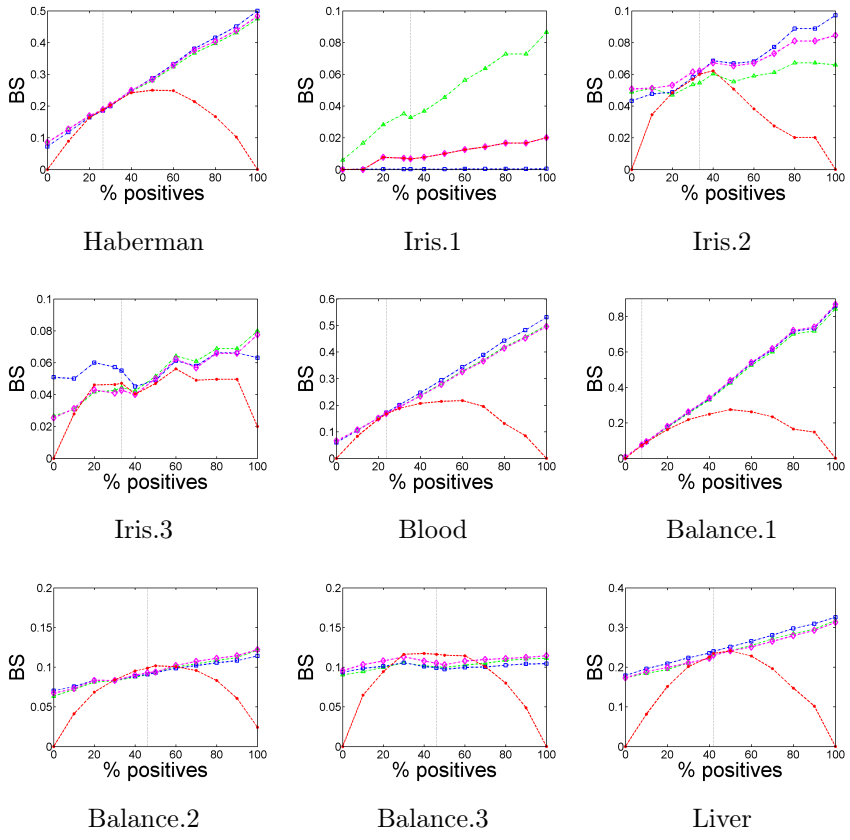
Urban.9



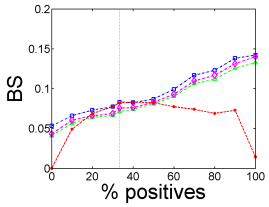
Musk

PCs per binary-class dataset

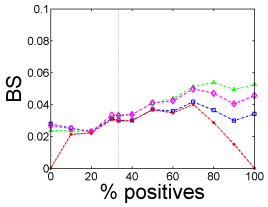
Appendix B: BS distribution per binary dataset by changing prevalence



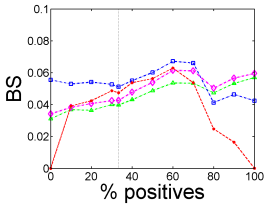
BS distribution by chainging prevalence (e.g., NB)



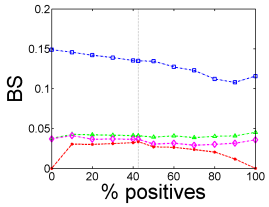
Seeds.1



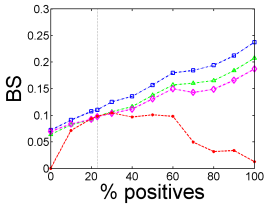
Seeds.2



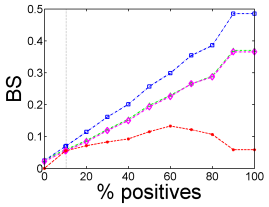
Seeds.3



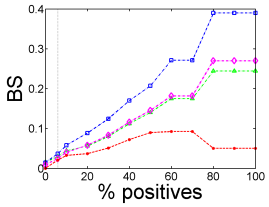
Ecoli.1



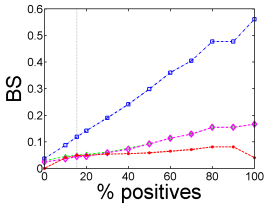
Ecoli.2



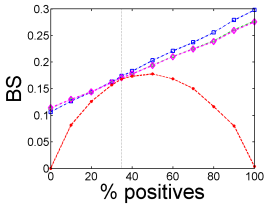
Ecoli.3



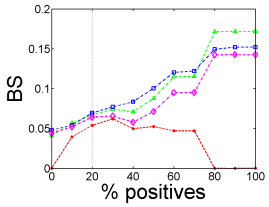
Ecoli.4



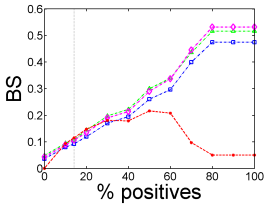
Ecoli.5



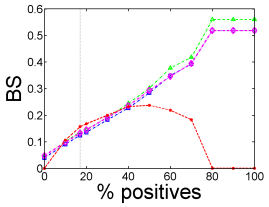
Pima



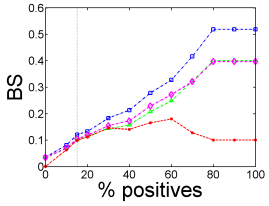
Breast.1



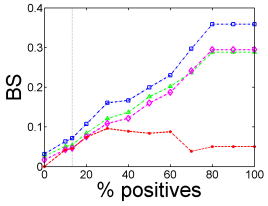
Breast.2



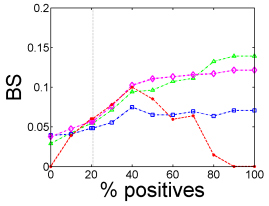
Breast.3



Breast.4

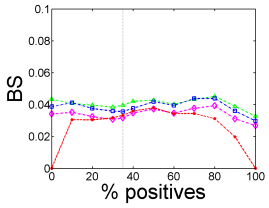


Breast.5

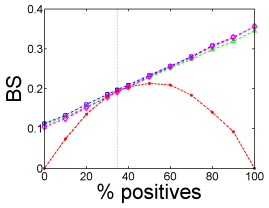


Breast.6

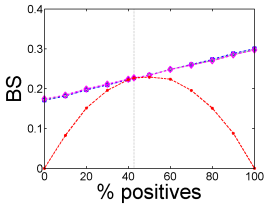
BS distribution by chainging prevalence (e.g., NB)



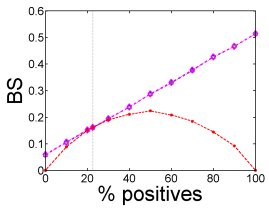
Wisconsin



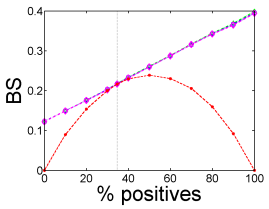
Tic-Tac-Toe



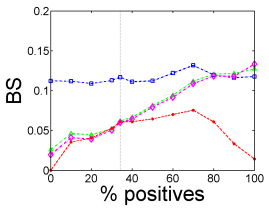
Contraceptive.1



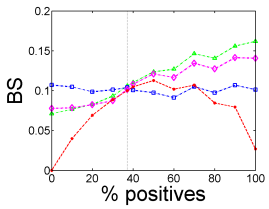
Contraceptive.2



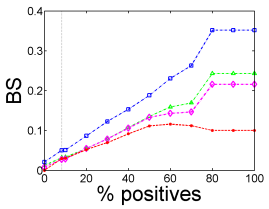
Contraceptive.3



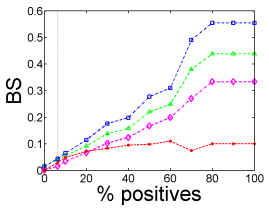
Glass.1



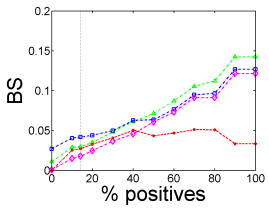
Glass.2



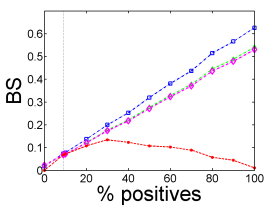
Glass.3



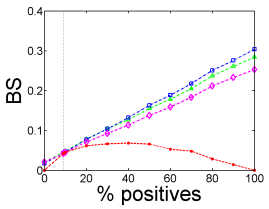
Glass.4



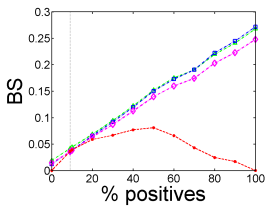
Glass.5



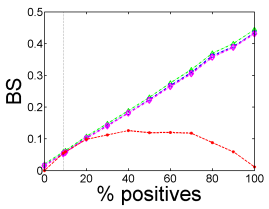
Vowel.1



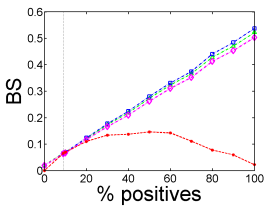
Vowel.2



Vowel.3

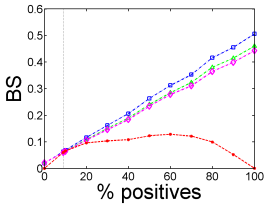


Vowel.4

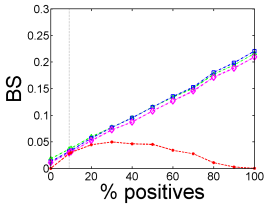


Vowel.5

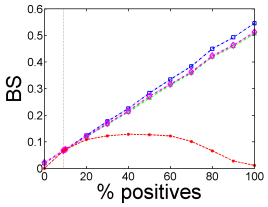
BS distribution by chainging prevalence (e.g., NB)



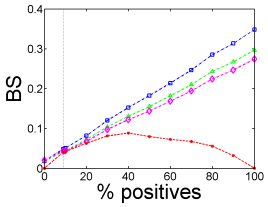
Vowel.6



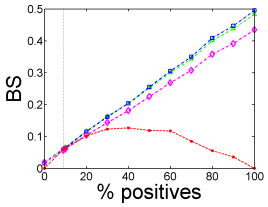
Vowel.7



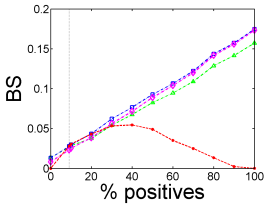
Vowel.8



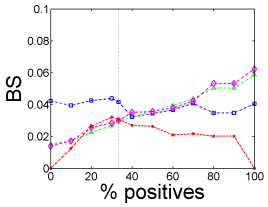
Vowel.9



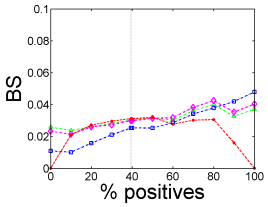
Vowel.10



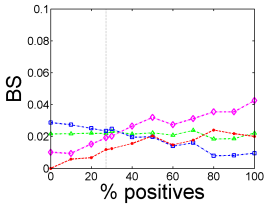
Vowel.11



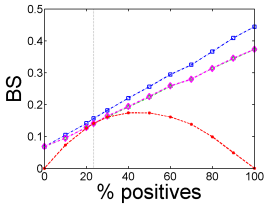
Wine.1



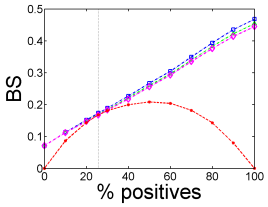
Wine.2



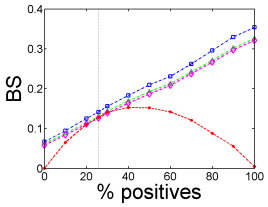
Wine.3



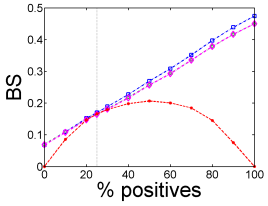
Statlog.1



Statlog.2



Statlog.3



Statlog.4

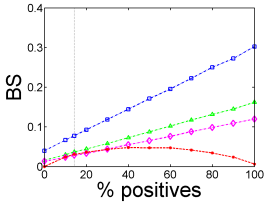


Image.1

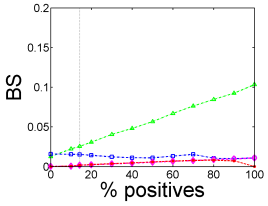


Image.2

BS distribution by chainging prevalence (e.g., NB)

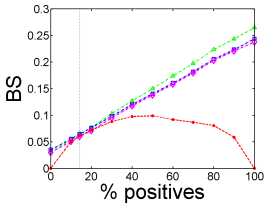


Image.3

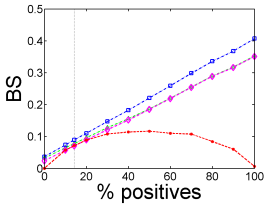


Image.4

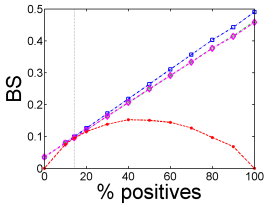


Image.5

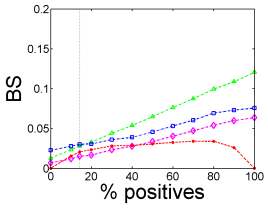


image.6

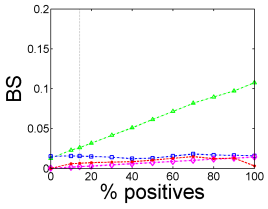
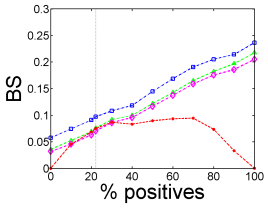
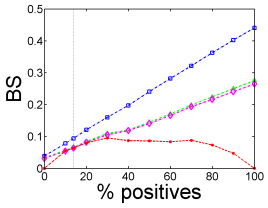


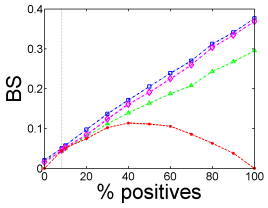
image.7



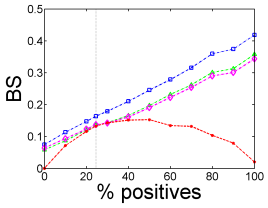
Cardio.1



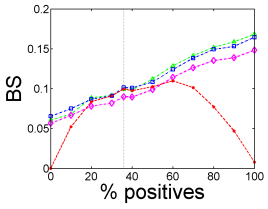
Cardio.2



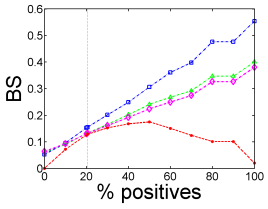
Cardio.3



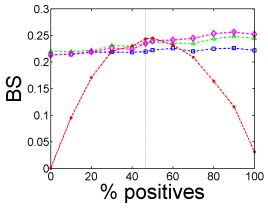
Parkinsons



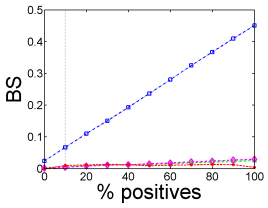
Ionosphere



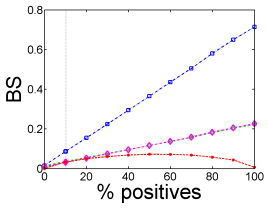
SPECTF



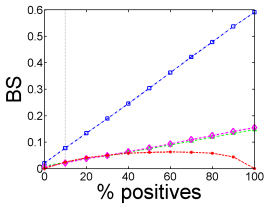
Sonar



Digits.1

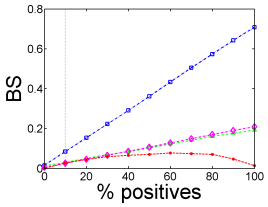


Digits.2

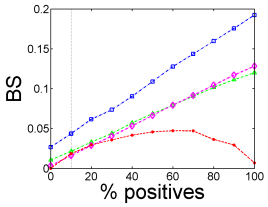


Digits.3

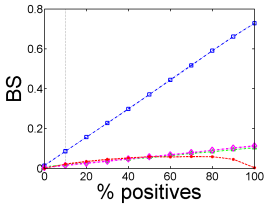
BS distribution by chainging prevalence (e.g., NB)



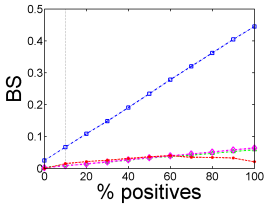
Digits.4



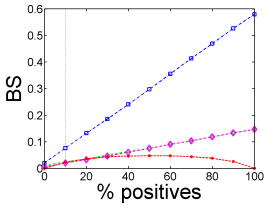
Digits.5



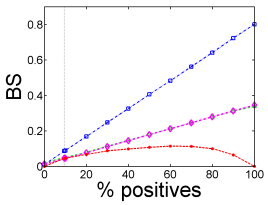
Digits.6



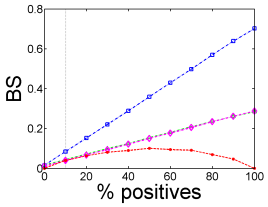
Digits.7



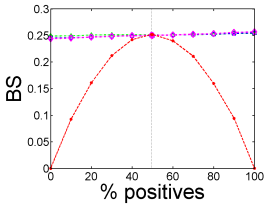
Digits.8



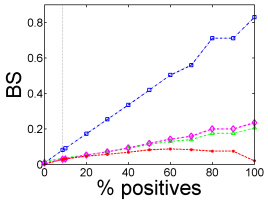
Digits.9



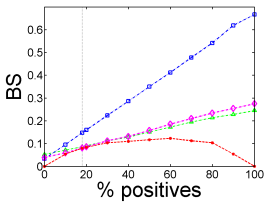
Digits.10



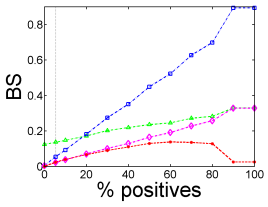
Hill-Valley



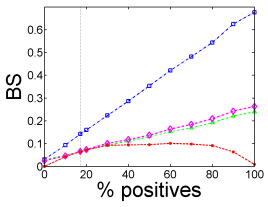
Urban.1



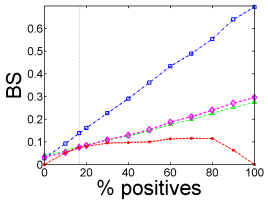
Urban.2



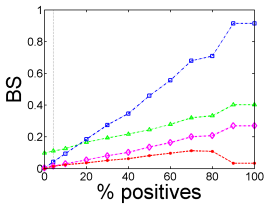
Urban.3



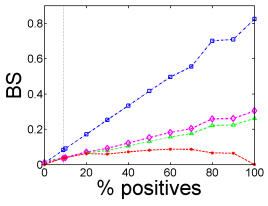
Urban.4



Urban.5

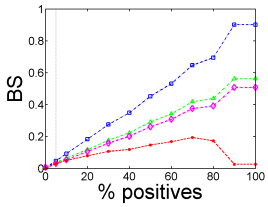


Urban.6

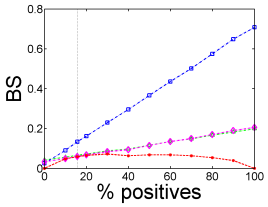


Urban.7

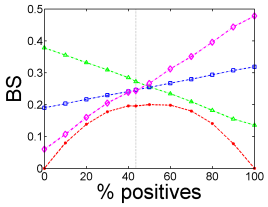
BS distribution by chainging prevalence (e.g., NB)



Urban.8



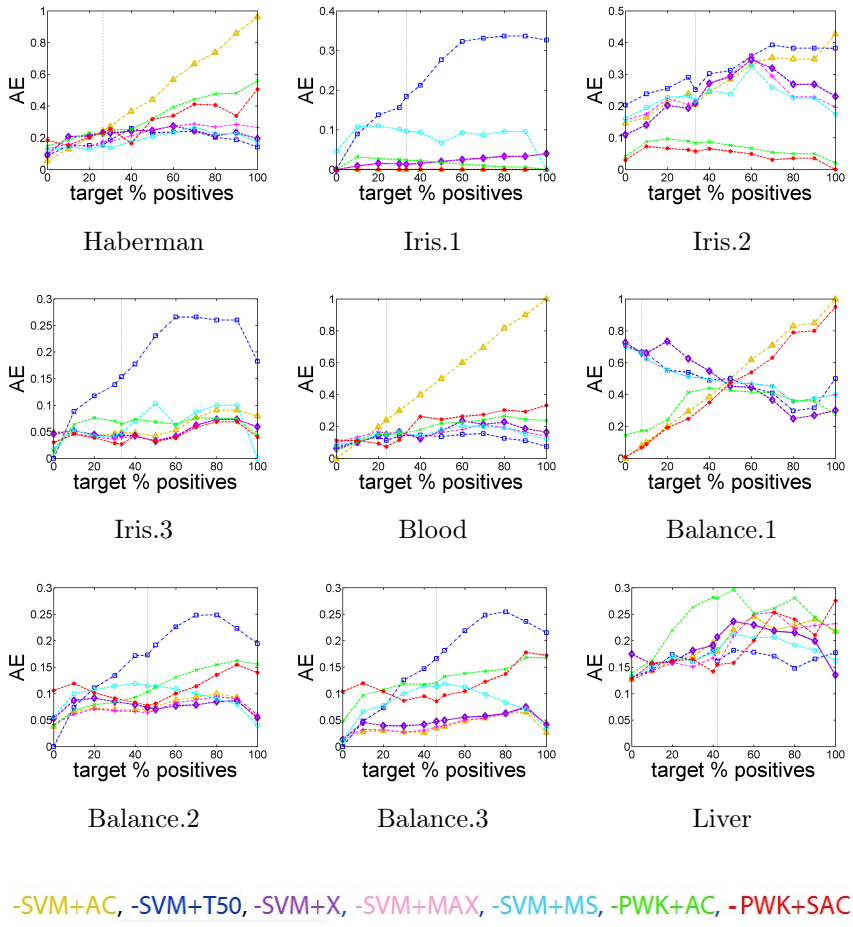
Urban.9



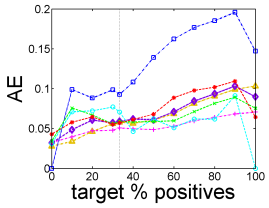
Musk

BS distribution by chainging prevalence (e.g., NB)

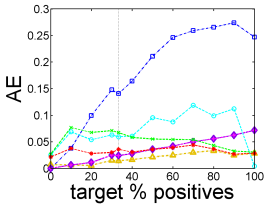
Appendix C: AE distribution per binary dataset by changing prevalence



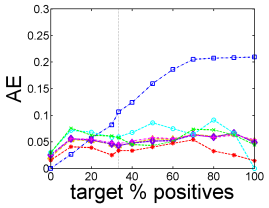
AE distribution by chainging prevalence



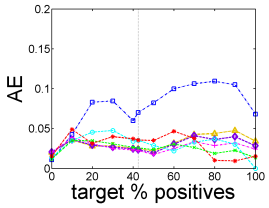
Seeds.1



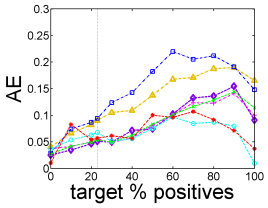
Seeds.2



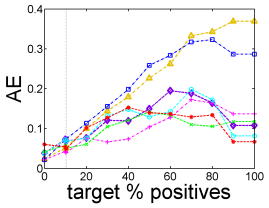
Seeds.3



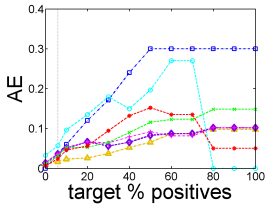
Ecoli.1



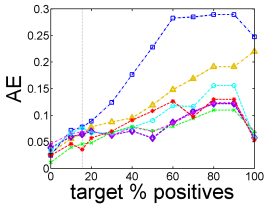
Ecoli.2



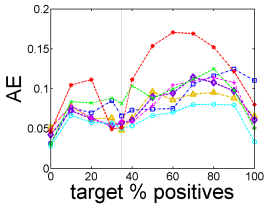
Ecoli.3



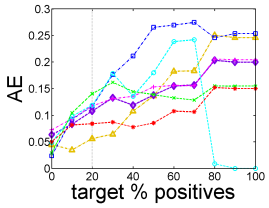
Ecoli.4



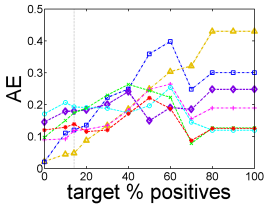
Ecoli.5



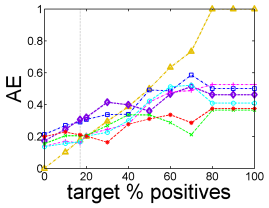
Pima



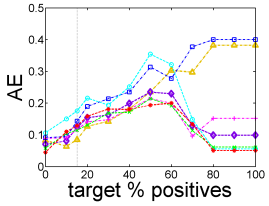
Breast.1



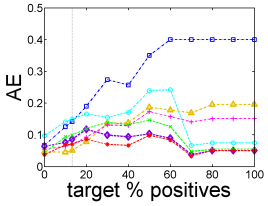
Breast.2



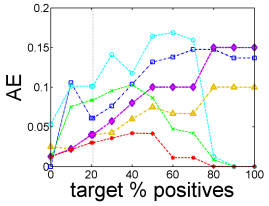
Breast.3



Breast.4

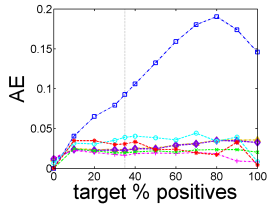


Breast.5

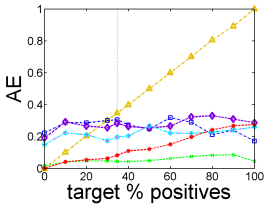


Breast.6

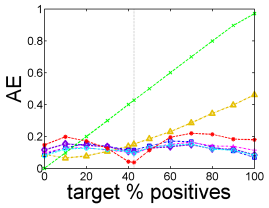
AE distribution by chainging prevalence



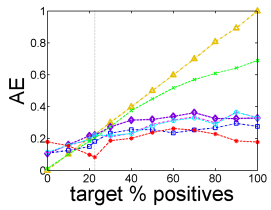
Wisconsin



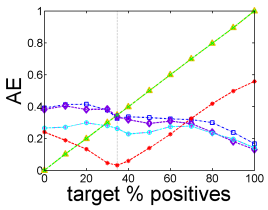
Tic-Tac-Toe



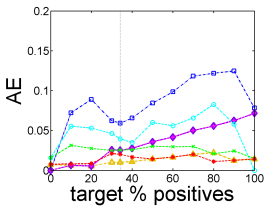
Contraceptive.1



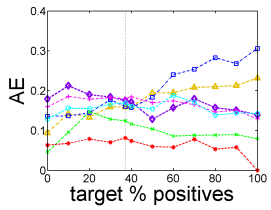
Contraceptive.2



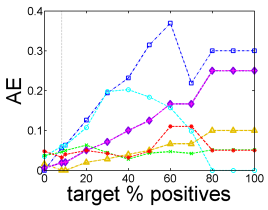
Contraceptive.3



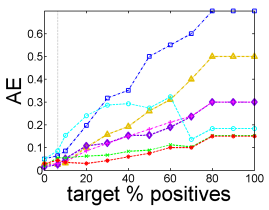
Glass.1



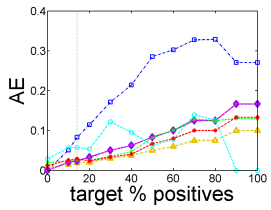
Glass.2



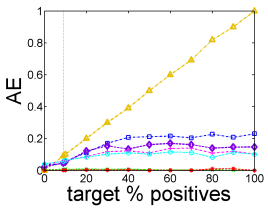
Glass.3



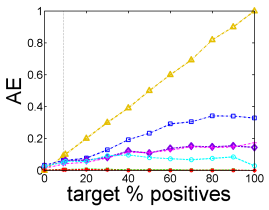
Glass.4



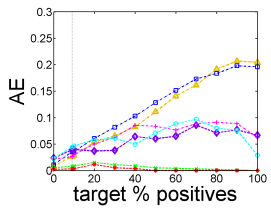
Glass.5



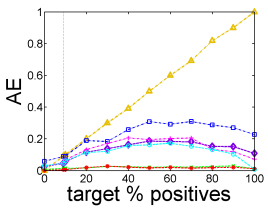
Vowel.1



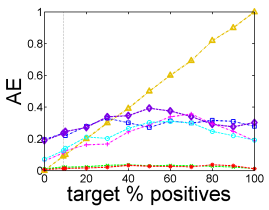
Vowel.2



Vowel.3

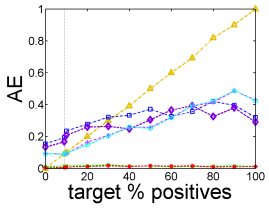


Vowel.4

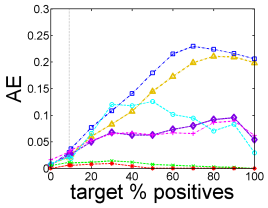


Vowel.5

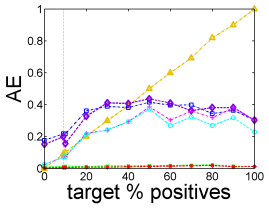
AE distribution by chainging prevalence



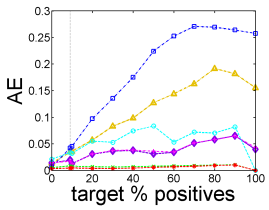
Vowel.6



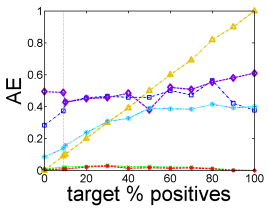
Vowel.7



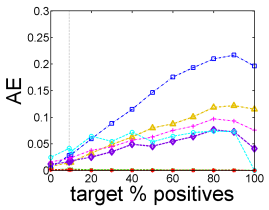
Vowel.8



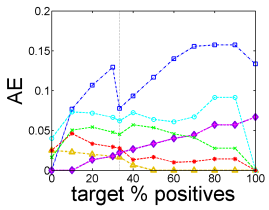
Vowel.9



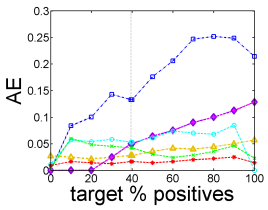
Vowel.10



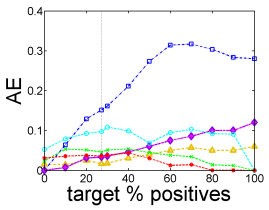
Vowel.11



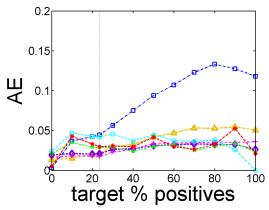
Wine.1



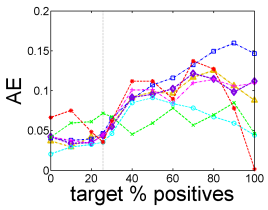
Wine.2



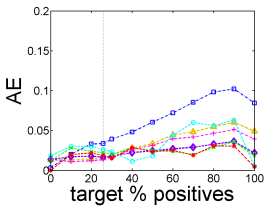
Wine.3



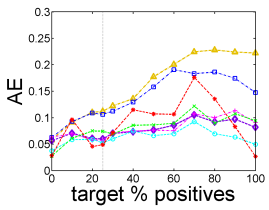
Statlog.1



Statlog.2



Statlog.3



Statlog.4

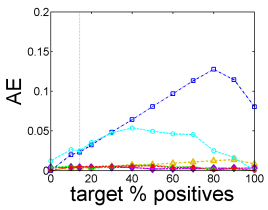


Image.1

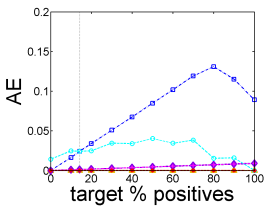


Image.2

AE distribution by chainging prevalence

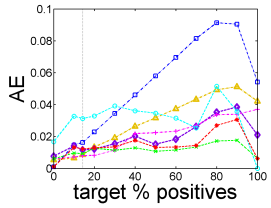


Image.3

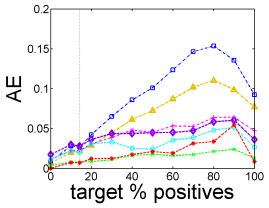


Image.4

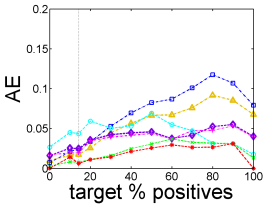


Image.5

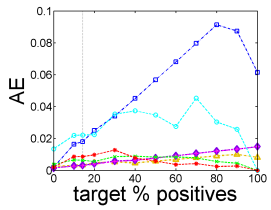


image.6

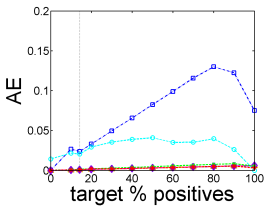
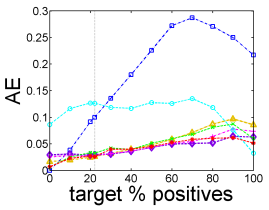
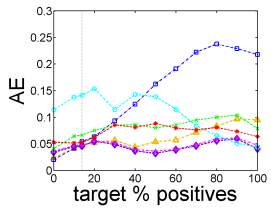


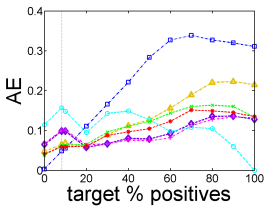
image.7



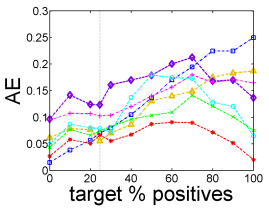
Cardio.1



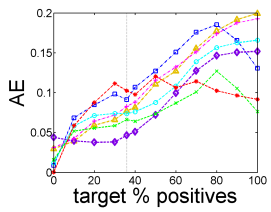
Cardio.2



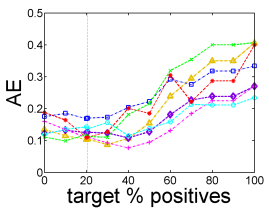
Cardio.3



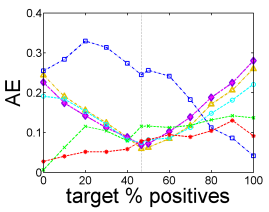
Parkinsons



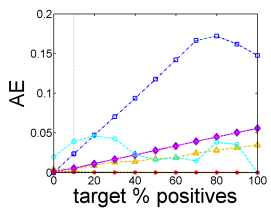
Ionosphere



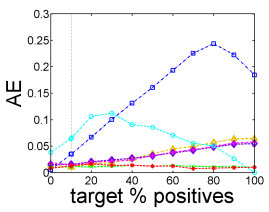
SPECTF



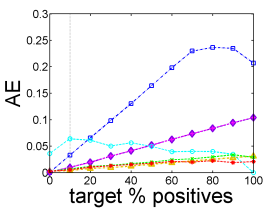
Sonar



Digits.1

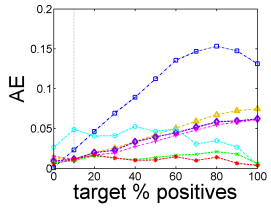


Digits.2

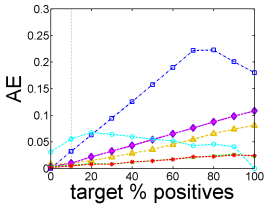


Digits.3

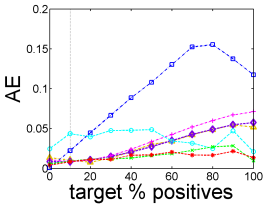
AE distribution by chainging prevalence



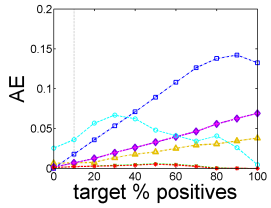
Digits.4



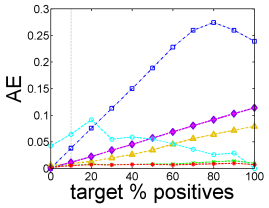
Digits.5



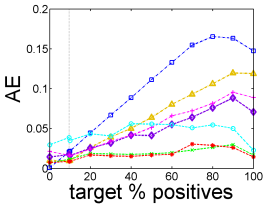
Digits.6



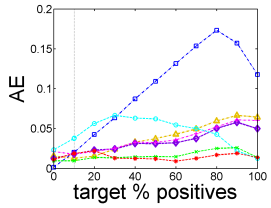
Digits.7



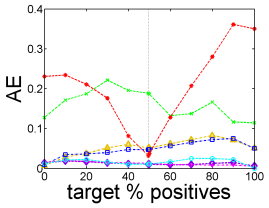
Digits.8



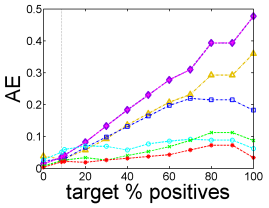
Digits.9



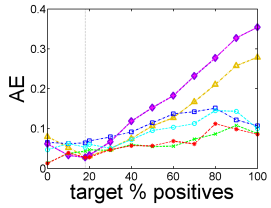
Digits.10



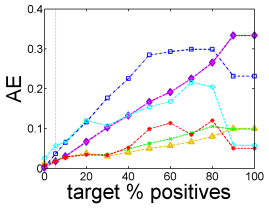
Hill-Valley



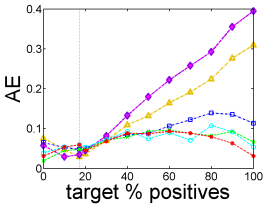
Urban.1



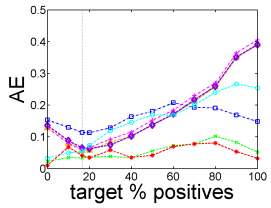
Urban.2



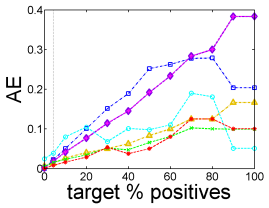
Urban.3



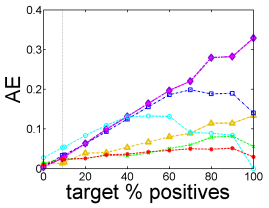
Urban.4



Urban.5

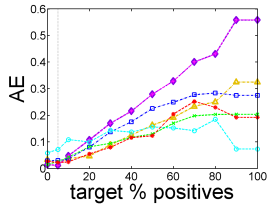


Urban.6

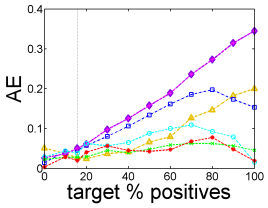


Urban.7

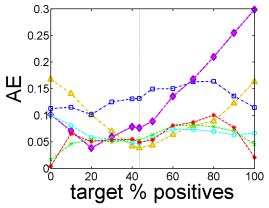
AE distribution by chainging prevalence



Urban.8



Urban.9



Musk

AE distribution by chainging prevalence

Appendix D: Application of SAC into MPI dataset

We applied the proposed SAC method into the MPI dataset to predict prevalence and conducted further analyses. Look at the Table below to see the various prevalence estimation methods' performances on the MPI dataset.

Summary of AE for all test conditions when applying MPI dataset

Measure	SVM					PWK	
	AC	T50	X	MAX	MS	AC	SAC
Average	0.470	0.109	0.100	0.152	0.144	0.102	0.141

We looked through the performance of each prevalence estimation method with respect to changes in target prevalence as shown in table below.

Summary of AE per test condition when applying MPI dataset

% positives	SVM					PWK	
	AC	T50	X	MAX	MS	AC	SAC
p(training)	0.130	0.078	0.053	0.104	0.104	0.069	0.022
0	0.000	0.066	0.050	0.117	0.108	0.034	0.084
10	0.100	0.088	0.059	0.120	0.120	0.065	0.011
20	0.200	0.089	0.092	0.067	0.067	0.082	0.088
30	0.300	0.083	0.084	0.143	0.146	0.116	0.192
40	0.400	0.145	0.116	0.153	0.153	0.122	0.171
50	0.500	0.109	0.113	0.158	0.140	0.142	0.114
60	0.599	0.113	0.096	0.204	0.195	0.123	0.164
70	0.700	0.144	0.141	0.207	0.199	0.127	0.248
80	0.805	0.132	0.128	0.187	0.170	0.139	0.218
90	0.906	0.138	0.159	0.194	0.167	0.121	0.190
100	1.000	0.122	0.113	0.171	0.156	0.079	0.190

As well, prevalence estimate result was applied as class prior for obtaining calibrated probability on the MPI dataset. Look at the Table below to see the various calibration methods' performances on MPI dataset.

Summary of BS for all test conditions when applying MPI dataset with estimated prevalence

Classfier	HB	PS	IR	RB	
				Actual Prev.	Estimated Prev.
NB	0.307	0.313	0.304	0.123	0.157
QDA	0.303	0.306	0.301	0.122	0.155
LDA	0.293	0.313	0.292	0.118	0.151
LR	0.290	0.309	0.290	0.120	0.153
SVM	0.331	0.364	0.333	0.134	0.169

We looked through the performance of each calibration method with respect to changes in test prevalence as shown in table below.

Summary of BS per test condition when applying MPI dataset with estimated prevalence (e.g., LDA)

% positives	HB	PS	IR	RB	
				Actual Prev.	Estimated Prev.
p(training)	0.099	0.103	0.099	0.099	0.100
0	0.025	0.023	0.025	0.000	0.013
10	0.082	0.085	0.082	0.080	0.081
20	0.139	0.147	0.139	0.134	0.143
30	0.196	0.208	0.196	0.168	0.203
40	0.253	0.270	0.252	0.187	0.224
50	0.311	0.333	0.310	0.193	0.219
60	0.367	0.394	0.365	0.186	0.221
70	0.425	0.456	0.423	0.168	0.232
80	0.484	0.519	0.482	0.130	0.200
90	0.542	0.581	0.540	0.074	0.125
100	0.597	0.640	0.594	0.000	0.056

국문초록

클래스 불균형과 클래스 오버랩 문제에서 분류모델의 클래스 예측값을 그대로 적용하는 것은 상당한 오류를 야기할 수 있다. 이는 클래스 불균형에서 대부분의 분류 알고리즘들이 실제보다 많은 인스턴스들을 다수 클래스 값으로 분류하며, 따라서 소수 클래스 값을 가지는 인스턴스들에 대해서는 잘못된 예측을 하기 쉽기 때문이다. 그리고 클래스 불균형에 일반적으로 수반되는 클래스 오버랩이 분류 문제에 대한 해결을 더욱 어렵게 만든다. 클래스 불균형과 오버랩 상황에서 발생하는 불완전한 분류작업은 클래스 값을 예측하는데 있어 많은 편향을 야기할 수 있는 있는데, 이때 소수 클래스 비율을 추정하는 분석적 방법론이 보다 적합한 접근이 될 수 있다. 왜냐하면 이러한 비율의 추정이 주어진 인스턴스가 소수클래스가 될 실제 가능성에 대한 보다 신뢰성 있는 정보를 제공하기 때문이다. 소수 클래스 비율을 추정하는 방법은 학습 데이터셋과 테스트 데이터셋간에 소수 클래스 비율이 크게 상이한 경우에도 테스트 데이터셋의 소수 클래스 비율을 정확하게 추정할 수 있을 때 그 타당성을 인정받을 수 있다. 소수 클래스 비율의 추정은 분할과 통합의 두 가지 측면으로 구분하여, 각각 (1) 보정된 확률값 획득과 (2) 전체 발생비율 예측으로 나누어질 수 있다. 우선, 본 연구는 테스트 데이터셋과 학습 데이터셋간에 소수 클래스 비율이 상이할 때에도 보정된 확률값을 정확하게 구하는 강건한 보정방법인 ROC(Receiver Operating Characteristics) Binning을 제안한다. 이는 클래스 발생비율 차이에 영향을 받지 않는 TPR(True Positive Rate)과 FPR(False Positive Rate) 을 사용하고 테스트 데이터셋에서의 전체 인스턴스들의 소수 클래스 발생비율을 직접적으로 반영한다. 이러한 방법은 ROC 그래프를 이용하여 특정 클래스 값 내 데이터 분포 특성과 클래스 발생비율의 영향을 분리함으로써 클래스

발생비율 차이에 강건한 성능을 가질 수 있다. 실험 결과, 테스트와 학습 데이터셋이 서로 다른 소수 클래스 발생비율을 가지고 있는 상황에서, 제안한 방법은 소수 클래스 발생비율 차이를 고려하지 않는 기존 방법에 비해 더 정확한 보정 확률을 제공하였다. 또한 제안한 방법의 보다 구체적인 활용 측면에서, ROC Binning과 이를 일부 조정한 TPR Binning이 한국군의 실제 MPI (Military Personality Inventory) 데이터에 적용된다. 한국군은 군면제가 요구되거나 군 입대 시 특별한 관심을 가지고 관리되어야 하는 군부적응병을 파악하는 것이 필요하다. MPI는 이러한 군부적응병을 예측하기 위해 사용된다. MPI는 대다수의 군인들이 큰 문제없이 만기 전역하며 일부는 부적응으로 인해 문제를 일으키는 일종의 클래스 불균형과 오버랩 문제로, 분류 모델의 성능이 좋지 않을 가능성이 높다. 그 대안으로 본 연구는 ROC Binning과 TPR Binning을 보정 확률, 다른 말로 유사한 MPI 테스트 결과를 공유하는 징병대상자들의 부적응 비율을 예측하기 위해 사용한다. 그리고 제안한 방법이 실제 MPI 데이터에서 높은 성능을 발휘하였는지 확인하고 실제 군 인사관리에서의 유용한 활용방법을 제안하였다. 두 번째로, 본 연구는 테스트와 학습 데이터셋 간에 소수 클래스 분포가 다를지라도 전체 소수 클래스 발생비율을 정확하게 구하는 간단하고 효과적인 예측방법인 SAC(Similarity Based Adjusted Count)를 제안한다. 이 방법은 AC(Adjusted Count) 방법을 기초로 하고 테스트 인스턴스들과 유사한 학습 인스턴스들의 TPR과 FPR을 이용한다. 일반적인 AC 방법은 TPR과 FPR 예측을 위해 전체 학습 데이터셋을 이용하나 제안하는 SAC는 학습 데이터셋 중 일부분을 주어진 테스트 데이터셋에 맞추어 이용한다. 테스트와 학습 데이터셋이 서로 다른 소수 클래스 비율을 가지고 있는 다양한 상황에서, 제안한 방법은 데이터 분포 차이를 고려하지 않는 기존 방법에 비해 더 정확한 발생비율 예측치를 제공하였다. 뿐만 아니라 본 연구는 SAC를 통해 예측된 발생비율을 베이지안 분류 모델을 활용해서 보정 확률을 구하고 인스턴스들을 분류하기 위해

필요한 클래스 사전 확률 정보로서 사용하여 그 효과성을 파악하였다. 베이지안 분류 알고리즘에서 클래스 값의 사전 확률은 분류결과에 직접적으로 영향을 미친다. 이 클래스 사전 확률은 본 연구의 클래스 발생비율과 본질적으로 동일하다. 테스트와 학습 데이터셋이 서로 다른 소수 클래스 비율을 가지고 있는 다양한 상황에서, 발생비율 예측치를 사전확률로 사용하는 것은 단지 학습 데이터셋의 클래스 발생비율에 의존하는 것보다 더 정확한 보정 확률과 클래스 예측치를 제공하였다. 부가적으로, 우리는 CGBN(correlation-based Gaussian Bayesian network)이라는 분류 정확도와 변수들간 상관관계를 모두 고려하는 베이지안 분류 알고리즘을 제안한다.

주요어: 데이터마이닝, 범주 불균형, 범주 오버랩, 소수 비율, 보정 확률, 전체 발생비율

학번: 2007-30167