Sampling Approaches for Unbalanced Data Classification Problem

---

A thesis submitted during 2011 to the University of Hyderabad in partial fulfillment of the award of a Ph.D degree in Computer Science.

by

Mrs. T. Maruthi Padmaja

Department of Computer and Information Sciences

School of Mathematics and Computer& Information Sciences

University of Hyderabad
(P.O.) Central University, Gachibowli
Hyderabad - 500 046
Andhra Pradesh
India

CERTIFICATE

This is to certify that the thesis entitled "**Sampling Approaches for Unbalanced Data Classification Problem**"submitted by **Mrs. T. Maruthi Padmaja** bearing Reg. No **06MCPC03** in partial fulfillment of the requirements for the award of Doctor of Philosophy in **Computer Science** is a bonafide work carried out by her under our supervision and guidance.

The thesis has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Prof. S. Bapi Raju
(Supervisor)

Dr. P. Radha Krishna
(Co-Supervisor)

Head of the Department

Dean of the School

DECLARATION

I **Mrs. T. Maruthi Padmaja** hereby declare that this thesis entitled "**Sampling Approaches for Unbalanced Data Classification Problem**"submitted by me under the guidance and supervision of Professor. **S. Bapi Raju** and **Dr. P.Radha Krishna** is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma.

Date: 17-6-2011 Name: T. Maruthi Padmaja

Signature of the Student:

Regd . No : 06MCPC03

ABSTRACT

Unbalanced data sets occur when one class (majority) of data severely outnumbers the target (minority) class. The performance of classification models learned by conventional algorithms such as Decision Tree, Neural Networks, and Naïve Bayes deteriorates due to unbalanced training sets. This thesis develops new resampling solutions for unbalanced training sets in the context of two-class classification problem, with emphasis on (i) overlapping class distributions and (ii) subspace preprocessing by dimensionality reduction. The main focus in developing new sampling solutions is to identify influence points from majority and minority class samples so that there will be clear discrimination between the two classes and the performance of the classification model can be improved.

Initially, a hybrid sampling criterion named extreme outlier elimination and hybrid sampling technique was devised to enhance the performance of Synthetic Minority Oversampling TEchnique (SMOTE) and Random UnderSampling (RUS). Since SMOTE projects new samples between minority class samples, outliers play a major role and cause degradation of classifier performance. On the other hand, removing all outliers (minority outliers), results in further loss of information of minority samples. So we have defined the extreme outlier concept and removed minority class outliers using k Reverse Nearest Neighbors (kRNN) method prior to the hybrid sampling of SMOTE + RUS. Proposed model is validated on publicly available Insurance Fraud Detection dataset using k-Nearest Neighbour, Radial Basis Function network, Decision Tree and Naïve Bayes classifiers and compared with SMOTE + RUS. Obtained results indicated that proposed method yielded better performance on all classifiers than rest of the other methods.

In order to counter the loss of majority class informative instances due to undersampling, a passive and an active sample selection criteria were proposed in the second part of the thesis. Proposed passive method, named Majority Filter based Minority Prediction (MFMP) comprises a two-step process. In the first step majority class points that are outside the minority class regions were selected through a minority class clustering process, and in the second step few more majority class samples were added through a random selection process. The final classification is carried out on the whole minority samples with selected majority class samples.

An active sample selection method, named, a Probabilistic Cost Weighted Statistical Query SVM (CstatQSVM) was proposed to select informative samples for Support Vector Machine classifier. Proposed CstatQSVM differs from existing active learning method Learning on the Border [39], in terms of informative sample selection, objective function and stopping criterion. Proposed MFMP as well as CstatQSVM achieved better classification performance than random undersampling which is quiet a popular solution for unbalanced training set problem.

In the last part of the thesis an investigation is carried out to identify, the effectiveness of PCA for preprocessing unbalanced datasets. Our investigation revealed that the directional difference between principle axes (PCs) of the two classes has an impact on the minority class prediction. Since PCA is unsupervised, it is not designed to use discriminative information to influence the directional difference between PCs corresponding to the classes. To alleviate this problem a class-specific $PC\_Synthetic\ Minority\ Oversampling\ Technique$ (CPC_SMOTE) was proposed. Proposed CPC_SMOTE defines the output subspace as a combination of class specific features extracted by applying PCA on individual class, with newly generated synthetic samples.

The viability of the proposed resampling methods was established using benchmark datasets from UCI [9] and UCD [38] machine learning repositories.

Kavithva varasini sagarabhyam, dourbhagya davambudha malikabhyam,
Dhoorikrutha namra vipathithabhyam, namo nama sri guru padukhabyam.
Anantha samsara samudhra thara naukayithabhyam guru
bhakthithabhyam, Vairagya samrajyadha poojanabhyam, namo nama sri
guru padukhabyam.

Dedicated To: *To my parents Ratna Kumari and Umakantha Rao*
To my lovable brother Dr. Anil Thuremella
To my Teachers
To my Guru Sri Sitarama swami and to Adi guru Lord Achyutha

Acknowledgments

My doctorial thesis is the longest and most difficult piece of work that I have ever undertaken and I am proud of it. At the same time I am aware of the fact that this achievement was not possible without people around me. This is a good time to look back and acknowledge the respective people.

It is my pleasure to thank Prof. S. Bapi Raju for supervising my Ph.D thesis at the University of Hyderabad. He is the key inspiration behind my research work. Another key personality behind this achievement is my second supervisor, Dr. P. Radha Krishna. My hearty acknowledgements for his great support and inspiration during the research work. Without regular feedbacks and recommendations from my two supervisors, Prof. Bapi Raju and Prof. Radha Krishna Sampling Approaches for Unbalanced Data Classification Problem would not have been possible.

My hearty acknowledgements to the Institute for Development and Research in Banking Technology (IDRBT) for financing my PhD thesis and for establishing interfaces with the University of Hyderabad.

My heartfelt gratitude to Dr. M.V.N.K Prasad for his moral support, encouragement and timely assistance in completing Ph.D during my days at IDRBT.

I wanted to thank Dr. Samba Murthy, Director, IDRBT, Prof. C. Raghavendra Rao, Head of the Department (DCIS), University of Hyderabad and Prof. T. Amaranath, Dean (School of MCIS), University of Hyderabad for providing necessary infrastructure and support.

Valuable suggestions from the Doctorial Review Committee (DRC) members helped to continuously enhance and polish my work. Therefore, I would like to thank my DRC members, Dr. Atul Negi and Dr. Rajeev Wankar. Special thanks to IDRBT review committee including Dr. V.N. Sastry, Dr. V. Ravi and Dr. A.R. Joshi.

My special thanks to Dr. V. Ravi , Dr. Arijit Laha and Rudra N. Hota for providing technical feedback on my Ph.D work, which was really helpful.

Supplementary facilities such as subsidized accommodation and timely conference registrations were helpful. Without subsidized accommodation, staying in

Hyderabad would be very difficult. Therefore, my sincere acknowledgements to the administration staff Pramod Kumar, Vijay Belugikar, T. Ashok Kumar, Dharmendar and Subrahmanyam. Furthermore, help from the librarian P.R. Kumar is highly acknowledgeable.

I learned working in the domain of research projects as a M. Tech student at the Tezpur University. Many thanks to all 2002-2004 faculty members at the department of Information Technology.

Efforts from M. Tech students at IDRBT that were involved in my project are acknowledgeable. Special thanks to Narendra Dhulipalla for the technical support and to Kasinath for his moral support.

Help from my friend and colleague, Kavitha Ammayappan is highly acknowledgeable. Therefore, I would like to thanks her hearty.

During my tough times counseling from Smt. Kiranmayi Bapi Raju helped me a lot. Many thanks for her efforts!

Suggestions from my senior researchers helped to plan a road map for my research. In particular, I want to thank Dr. Pradeep Kumar and Dr. Geeta Kumari for their valuable suggestions. Special thanks to my friends including Dr. Vijay Kumar, Dr. Suresh Babu and Renuka Methre. Regular informal discussions with them helped to enhance my approaches.

Thanks to my M.Tech class boys during 2002-2004 at Tezpur university, who really helped a lot during my studies at Tezu as a single girl in the 15 member class, which is a step in to take up this research project.

Last but not least I would like to thank M.Tech students at IDRBT during my period Nikunj Chowhan, Hanumantha Rao and Ramu Kunta, G. Raghu, Srinivas and Janaki Saran.

Within this restricted space, I would not be able to thank all stakeholders involved in my achievement. Therefore, many thanks to all people that were directly and indirectly involved in my work.

T. Maruthi Padmaja

Contents

1 Introduction **1**

1.1 Unbalanced Data Classification 1

1.2 Problem Statement 4

1.3 Contributions 5

1.4 Organization of The Thesis 8

2 Related Work **11**

2.1 Effect of Unbalanced Datasets on Classifier Performance 11

2.2 Methods for Handling Unbalanced Datasets 13

2.2.1 Data Level Solutions 13

2.2.2 Algorithm Level Solutions 20

2.2.3 Contributions from this thesis 27

2.2.4 Chapter Summary 27

3 Preliminaries **30**

3.1 Machine Learning Classification Algorithms 30

3.1.1 Decision tree (DT): 30

3.1.2 Naïve Bayes (NB): 31

3.1.3 k-Nearest Neighbour (kNN): 31

3.1.4 Radial Basis Function Networks (RBF): 32

3.1.5 Support Vector Machine (SVM): 33

3.2 Performance Measures 34

3.2.1 Performance Measures on Unbalanced Distributions 35

3.3 Comparison of Performance of Classifiers 37

3.3.1 Paired T-Test 37

3.3.2 Wilcoxon Signed-Ranks Test 37

3.3.3 Friedman's Test 38

3.4 Chapter Summary 39

4 Extreme Outlier Elimination and Sampling Techniques 41

4.1 Introduction 41
4.2 Background 43
4.2.1 Hybrid of Synthetic Minority Oversampling Technique and RUS 43
4.2.2 Outlier Detection and Filtering by RNN 45
4.3 Extreme Outlier Elimination using kRNNs + Hybrid Sampling . . . 46
4.4 Case study with Insurance Fraud Dataset 49
4.4.1 Related Work on Fraud Detection 49
4.4.2 Dataset Description 50
4.4.3 Value Difference Metric 51
4.4.4 Experimental Results and Discussion 51
4.5 Chapter Summary 58

5 Majority Filter based Minority Prediction: (MFMP) 61

5.1 Introduction 61
5.2 Background 62
5.2.1 Partition Around Medoid (PAM) Clustering Algorithm . . . 62
5.3 Majority Filter-Based Minority Prediction (MFMP) 63
5.4 *RNN* Curve based Cluster Counting 67
5.4.1 Generating Reverse-NN Curve 67
5.4.2 Counting Clusters in Reverse-NN Curve 68
5.5 Experimental results 72
5.6 Chapter Summary 77

6 A Probabilistic Cost Weighted Active Learning Approach 78

6.1 Introduction 78
6.2 Related Work 80
6.3 Background and Motivation 80
6.3.1 Different Error Cost (DEC) SVM 81
6.3.2 Active Learning 81
6.4 Exploratory Study of StatQSVM 84
6.4.1 Datasets 84
6.4.2 Discussion on StatQSVM 86

6.5 Proposed Algorithm . . . 87
6.5.1 CStatQSVM Algorithm . . . 87
6.5.2 Stopping Criterion . . . 89
6.6 Empirical Evaluation of CStatQSVM . . . 90
6.6.1 Discussion on CStatQSVM . . . 90
6.7 Chapter Summary . . . 97

7 Is PCA Effective for Preprocessing Unbalanced Data? 101
7.1 Introduction . . . 101
7.2 Related Work . . . 102
7.3 PCA and its Possible Effects . . . 104
7.4 Evaluating PCA Performance over Unbalanced Datasets . . . 106
7.5 Experiments on Synthetic Datasets . . . 107
7.5.1 Experiment-A . . . 108
7.5.2 Experiment-B . . . 112
7.6 Experiments on Real World Datasets . . . 116
7.6.1 Datasets . . . 116
7.6.2 Experimental Results and Discussion . . . 116
7.7 Chapter Summary . . . 122

8 CPC_SMOTE 127
8.1 Introduction . . . 127
8.2 Related Work . . . 128
8.3 The CPC_SMOTE Framework . . . 129
8.4 Experimental Evaluation . . . 132
8.4.1 Evaluation Metric . . . 132
8.4.2 Datasets . . . 132
8.4.3 Experimental Results and Discussion . . . 135
8.5 Chapter Summary . . . 138

9 Conclusion and Future Work 139
9.1 Conclusions . . . 139
9.2 Future Work . . . 141

A Description of Insurance Fraud Dataset 160
A.1 Data Description . . . 160

A.2 Data Quality 161
A.3 Data Preparation 163
A.3.1 Select the data 163
A.3.2 Construct the data 163

B Description of Real-World Datasets **166**

C Description of Synthetic Data Sets used **170**
C.0.3 Synthetic Dataset-1 170
C.0.4 Synthetic Dataset-2 170
C.0.5 Synthetic Dataset-3 171
C.0.6 Synthetic Dataset-4 171

D Description of Other Performance Measures **174**

List of Figures

1.1 Classification of Unbalanced data 3

2.1 Taxonomy of research solutions to address class imbalance problem 14
2.2 Three level architecture for multiple resampling 19
2.3 Roadmap of the proposed solutions in unbalanced data taxonamy . 28

3.1 Architecture of Radial Basis Function Network 32
3.2 Training a Support Vector Machine 33

4.1 Synthatic sample generation in SMOTE 42
4.2 Generation of samples for training using $kRNN$ 48
4.3 TP_{rate} Vs OFD rate for DT 53
4.4 TN_{rate} Vs OFD rate for DT 53
4.5 TP_{rate} Vs OFD rate for NB 54
4.6 TN_{rate} Vs OFD rate for NB 54
4.7 TP_{rate} Vs OFD rate for kNN 54
4.8 TN_{rate} Vs OFD rate for kNN 54
4.9 TP_{rate} Vs OFD rate for RBF 54
4.10 TN_{rate} Vs OFD rate for RBF 54

5.1 Flow diagram for MFMP. 65
5.2 A query point located in a dense region of the data space 71
5.3 A query point in a sparse region of the data space 71
5.4 Results of Decision Tree Classifier 76
5.5 Results of Naïve Bayes Classifier 76
5.6 Results of k-Nearest Neighbour Classifier 76
5.7 Results of Radial Basis Function Network Classifier 76

6.1 Sample selection criterion for StatQSVM 83

6.2 Comparison across StatQSVM and LOB 89
6.3 Stopping criterion for CStatQSVM algorithm 92
6.4 Comparison of CStatQSVM performance with StatQSVM and LOB 99

7.1 Schematic diagram for the sample generation in angular separation. 107
7.2 Reprojection error for angular separations 108
7.3 $F-measure$s from kNN classifier 109
7.4 $F-measure$ from PCA+kNN classifier 109
7.5 Schematic diagram for generating samples with varied degree of overlapping.112
7.6 Reprojection error with varied degree of overlapping 113
7.7 Minority class prediction over different degrees of overlapping and IR115
7.8 Block diagram showing combination of experiments 118

8.1 Flow diagram for CPC_SMOTE framework. 130
8.2 Data from classes ω_1 and ω_2 134

9.1 Roadmap of the proposed solutions in unbalanced data taxonamy . 141

C.1 Schematic diagram for the sample generation in angular separation. 171
C.2 Schematic diagram for generating samples with varied degree of overlapping.172
C.3 Data from classes ω_1 and ω_2 173

D.1 An ROC graph showing performance for two different classifiers C_1 and $C_2$175
D.2 The difference between comparing algorithms in ROC vs PR space . 176
D.3 Mapping of ROC points to Cost lines 178
D.4 Comparison of ROC curves and Cost curves 178

List of Tables

1.1 Dataset Complexities[19, 63, 65, 132] 2

2.1 Cost Matrix 25

3.1 Confusion Matrix 34

4.1 Notations and definitions used in this chapter 45
4.2 Comparison of Method-a with Method-b 53
4.3 Comparison of $G - mean$ across Method-a and Method-b 55
4.4 Test set results across Original Data (OD), RUS and ROS 57
4.5 Ranking of classifiers based on $G - mean$ across OD, RUS and ROS 59

5.1 Dataset Description 74

6.1 Datasets Description 85
6.2 Comparison of Average F-measure and training time 94
6.3 Ranking of minority class *F-measure* across different methods . . . 96

7.1 Dataset Descriptions 117
7.2 kNN classification results on original input space and PCA subspace120
7.3 Decision tree classification results on original input space and PCA subspace124
7.4 Naïve Bayes classification results on original input and PCA subspace125
7.5 Cosine angle values between minority class and majority class PC's 126

8.1 Datasets Description 134
8.2 Comparison of CPC_SMOTE performance with Original Data, PCA, SMOTE136

A.1 Original attributes in automobile insurance fraud dataset. 162
A.2 Modified attributes in the dataset. 165

B.1 Characteristics of Unbalanced Datasets Investigated 169

D.1 Performance measures and usage 177

List of Algorithms

1 Psuedocode for AdaBoost algorithm 23
2 Pseudocode for SMOTE Algorithm 44
3 PAM Clustering Algorithm 63
4 Proposed MFMP Algorithm 66
5 Algorithm for RNN Curve generation 69
6 Algorithm for finding cluster count in RNN curve 70
7 Algorithm for finding the length of RNN curve 72
8 MFMP Algorithm based on automatic cluster counting using RNN 73
9 Pseudocode for CStatQSVM algorithm 91

Chapter 1

Introduction

This chapter introduces the problem of classification of unbalanced data and provides a glimpse of proposed solutions. Further, this chapter also presents the organization of the dissertation work.

1.1 Unbalanced Data Classification

Classification algorithms utilize supervised learning method and learn a model from already labeled historical data and use this model to predict the class labels of unseen test data. Most widely used classification algorithms are Decision Trees (DT), Neural Networks (NN), Support Vector Machines (SVM), k-Nearest Neighbours (kNN), and Naïve Bayes (NB). Each of these classifiers has different learning capabilities and distinct learning biases [85]. Recent research focuses on unbalanced training sets in classification problem. The unbalanced class distribution problem occurs in a training set when one class (majority) of data severely outnumbers the target (minority) class. As a consequence, the performance of the existing classification algorithms tends to be biased towards the majority class. The fundamental issue with the existing classification algorithms is that the assumption of balanced class distribution in training set and/or the assumption of equal misclassification costs associated with each of the classes. However, these assumptions are not appropriate for real world situations. There is a wide range of domains, where misclassification can be costly for minority classes than majority class. Such domains are [56, 63, 111]

- Network Intrusion detection

- Fraud detection
- Detection of oil spills using radar images of the ocean surface
- Helicopter gear-box fault monitoring
- Earthquake and nuclear explosions
- Identifying defects in modern manufacturing plants
- Text classification

Apart from the learning assumptions of classification algorithms, the dataset complexities (Table 1.1) which include class complexity, overlapping classes, size of the dataset and disjuncts also play a crucial role in performance deterioration in case of unbalanced datasets [19, 63, 65, 132].

Table 1.1: Dataset Complexities[19, 63, 65, 132]

Dataset complexities	Description
Class Complexity	The number of training samples required to represent.
Overlapping classes	Deals with the degree to which the minority class,overlaps with majority class. Linearly separable domains are not sensitive to any amount of imbalance.
Dataset Size	Large training sets are seen to be less sensitive towards class imbalance, for instance the training set with size 900:100 is less sensitive to imbalance than the training set with 90:10 size. General classifier on the latter training set experiences lack of information to distinguish minority class.
Disjuncts	In concept learning, disjunct is a conjunctive definition that describes the subconcept of original concept. In case of unbalanced datasets, the minority samples are prone to forming small disjuncts that cover only few samples and have more error rate compared to majority class disjuncts.

Based on the minority data complexity, two kinds of unbalanced data classification problems can occur, one is because of relative lack of information and

the other is because of absolute lack of information [133]. Figure 1.1 depicts the flow diagram for unbalanced data classification problems and the corresponding classifiers that suffer from one or both of the of the two problems.

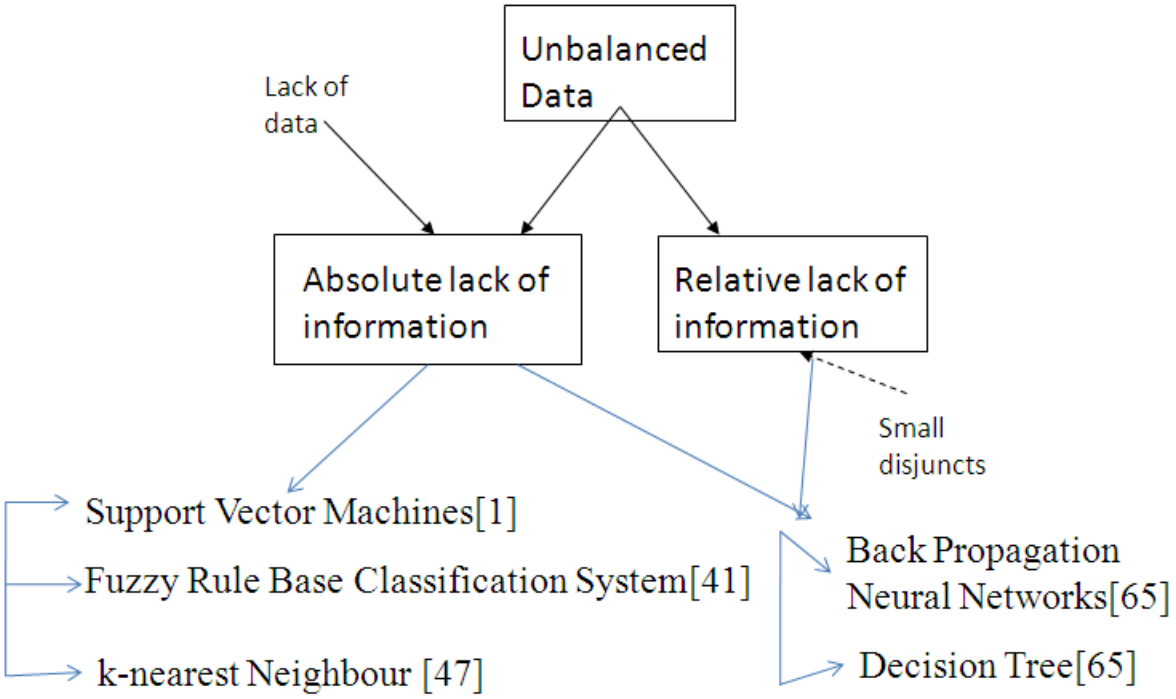


Figure 1.1: Classification of Unbalanced data

Several researchers analyzed the effect of unbalanced datasets on various classification algorithms [1, 5, 41, 47, 63, 141] and compare their performance one against other. Their studies indicated that Support vector Machines, k-Nearest Neighbour and Fuzzy Classifier Systems are less sensitive to unbalanced class distributions when compared to Decision Tree and Neural Networks[41, 47, 63] classifiers. Fig. 1.1 depicts the scenario that DT and NN suffer from both absolute lack and relative lack of information [19, 65, 131, 132], whereas the SVM, $FRBCS$, kNN suffer from absolute lack of information [1, 41, 47]. But recently proposed decision tree induction using splitting criteria like Hellinger Distance [26] and Class Confidence Proportion (CCP) measure [78] enable DT to be skew invariant for unbalanced class distributions.

In literature, to counter the bias caused by unbalanced datasets, several resampling solutions were discussed in [63] to balance the class distributions. Resampling solutions include oversampling (increasing the size of minority class distribution by replication or by informative generation [18]) and undersamling (decreasing

the size of majority class distribution by randomly rejecting or by informatively selecting [150] few majority class samples). But incorporating additional minority class data in terms of oversampling can cause computational overhead. Removing majority class samples in terms of undersampling can lead to loss of potential information from majority class [1, 35].

Apart from resampling techniques, cost-sensitive learning was proposed to handle unbalanced class distributions [34, 71]. These methods assume different misclassification costs for each class and for unbalanced datasets misclassification cost for the minority class is assigned to a higher value than that of the majority class. However for cost sensitive learning, tuning of misclassification costs is required in cost matrix, if actual misclassification costs are unknown. Another line of research [62, 99] suggested that novelty detection approaches such as one-class SVM and one-class neural networks can be effective than discriminative learning techniques in order to handle highly unbalanced datasets. However, the success of one-class classification methods rely on tuning of the threshold imposed on target class boundary. Too strict threshold can aggravate minority class prediction by leaving the minority samples outside the boundary whereas too loose threshold can aggravate majority class prediction by incorporating the majority samples inside the boundary. However, in handling unbalanced datasets, discriminative learning algorithms proved to be more efficient than one-class learning algorithms. However, the studies in [81, 130] indicated that resampling using undersampling and oversampling can be of the same effect as the tuning of cost matrix as well as adjusting the decision threshold in the Receiver operating characteristics (ROC) curve.

This thesis aims to develop more focused resampling methods for improving the two-class classification performance on overlapped and subspace preprocessed training sets. In this thesis the subspace preprocessing is considered by dimensionality reduction.

1.2 Problem Statement

The main objective of this thesis is to develop new sampling solutions for the challenging scenario of learning classification models from the unbalanced two-class training distributions with an emphasis on

1. Overlapping classes

2. Subspace preprocessing

As per our knowledge, resampling solutions proposed so far improve the minority class prediction with a great trade-off between minority class prediction rate and majority class prediction rate. Increasing minority class prediction rate at the same time without losing the prediction from majority class is an important issue.

This thesis develops new informative resampling solutions to maintain the trade-off between majority and minority class prediction in classifier performance.

Despite the bias caused by majority class and data complexity, the dimensionality of the dataset also plays a crucial role in classifier performance. As the number of dimensions increases, the classification performance drops-down. In order to improve the performance of the classifier over high dimensional datasets, the data mining practitioners generally use Principal Component Analysis (PCA). PCA is a global dimensionality reduction technique based on the principle of maximizing the variance with minimum reconstruction error [66]. Since PCA is unsupervised, this thesis investigates whether PCA is adequate enough as a preprocessing tool to hold minority class discriminative information for unbalanced class distributions and proposes an appropriate solution based on class-specific principal component analysis. On a broader view, this thesis addresses the general problem of handling unbalanced two-class training sets with an emphasis on the following problems,

- Enhancing the classifier performance while oversampling
- Enriching the classifier performance while undersampling
- Enriching SVM classifier performance while undersampling
- Investigating the effectiveness of PCA for preprocessing unbalanced data in the context of two-class classification problem
- Enhancing the classifier performance over PCA subspace of unbalanced datasets

1.3 Contributions

We summarize the main contributions of the thesis below:

(a) **Improved hybrid oversampling method**

As an initial step, we have validated the hybrid of Synthetic Minority Oversampling TEchnique + Random undersampling (SMOTE + RUS) on Insurance fraud detection dataset in order to improve Fraud Detection rate. SMOTE is a guided approach to choose additional minority samples. However, due to its nature of projection of samples between two minority samples, outliers will play major role and deteriorate classifier performance. At the same time, removing minority outliers causes loss of minority class which has fewer samples information. So, in this work, we defined *extreme outlier* concept and identified the the extreme outliers to eliminate them from SMOTEing. To find the extreme outliers, we used k-Reverse Nearest Neighbour ($kRNN$) algorithm. $kRNN$ algorithm chooses a sample which is a part of kNN of other samples.

To validate our approach, experiments were carried out on global classifiers namely DT, RBF, NB and on local classifiers such as kNN. The results also were compared with other approaches namely SMOTE+RUS, Random undersampling (RUS) alone and with random oversampling (ROS). Experimental evidence shows that ignoring minority class extreme points with hybrid sampling can improve the minority class prediction rate (TP rate) and majority class prediction rate (TN rate), thus improving the overall classifier performance.

(b) **Designing a new passive sample selection method for improving the classifier performance**

According to Drummond and Holte [35], random undersampling enhances the minority class prediction, but it also leads to loss of potential majority class information. We proposed a new undersampling method, named, Majority Filter-based Minority Prediction (MFMP) to improve the classifier performance. It comprises a two-step process. Let S be the bin of whole training set and S_{min} be the bin for minority class samples. As a first step the minority samples are grouped by clustering process and the majority class samples falling within these minority clusters are added to S_{min} and a classifier is learned on S_{min}. As a second step, from each minority class cluster whose imbalance ratio is greater than equal to 50%, majority class samples are randomly selected and added to S until there is an improvement in minority class performance. Since minority samples are less in number and sparse, it is difficult to decide the number of clusters. In this work, we designed a novel cluster counting approach by developing a Reverse Nearest Neighbour (RNN) curve to determine the number of potential clusters before

actually clustering the minority samples.

Experimental evidence on one synthetic and 3 UCI repository datasets over DT, RBF, NB and kNN classifiers, clearly indicated that informatively selecting majority class samples leads to superior performance than randomly undersampling the majority class on all considered classifiers.

(c) **Introducing a fast active sample selection method to select informative instances for Support Vector Machine classifier**

Due to the loss of informative instances from majority class, undersampling greatly affects the orientation of hyperplane of SVM classification model. However, active learning on SVM model enables to query the informative samples at the time of learning itself. In this work, we developed a probabilistic cost weighted active learning approach (CStatQSVM) to address the above problem. CStatQSVM is an undersampling method, which informatively selects instances from both classes based on a confidence factor. Here, Different Error Cost (DEC) [126] was used for deriving costs at every iteration of probabilistic querying process. Proposed algorithm is characterized with a new stopping criterion based on confidence factor stabilization over different error costs. We evaluated the CstatQSVM algorithm on 9 UCI repository benchmark datasets and compared with Learning on the Border active learning method and other conventional methods that address the class imbalance problem. Results demonstrate that CstatQSVM improves the minority class performance than random undersampling, DEC and LOB. Moreover CSTATQSVM is faster than LOB.

(d) **Is PCA effective for preprocessing unbalanced data?**

It has been observed that applying classical PCA on unbalanced datasets can not significantly remove the redundant information. In this work, we have analyzed the role of PCA on unbalanced classification problems with classifiers such as DT, kNN and NB classifiers. Empirically we have studied the effect of PCA on two sets of alignments of principal axes (PCs) on synthetically simulated datasets. The results obtained on simulated datasets are further validated over 10 benchmark datasets. The results drawn on simulated as well as real datasets indicated that the directional difference between the principal axes corresponding to two classes can lead to loss of discriminative information from minority class in terms of reconstruction error as well as deterioration of the minority class prediction. Reconstruction error root mean square error (RMSE) and angular separation be-

tween two class PCs were used for measuring the tendencies of minority class in PCA subspace. We propose that the angular separation between the PC's can be a useful metric for the viability of PCA based preprocessing on unbalanced training sets.

(e) **Devising a class-specific principal component based resampling method to alleviate the bias caused by majority class variance.**

This work proposes a class-specific principal component based resampling method, called, CPC_SMOTE, which combines class-specific features extracted by applying PCA on each class. Later synthetic samples are generated over the projected data of the combined subspace. Proposed CPC_SMOTE is compared with PCA and SMOTE algorithms. The performance of proposed model is evaluated using classification accuracy and minority class $F - measure$. Obtained results on one synthetic and 4 datasets from UCI as well as UCD repositories indicated that CPC_SMOTE yields superior classifier performance on different unbalanced datasets where the maximum variance predominantly represents majority class.

1.4 Organization of The Thesis

The thesis focuses on developing new resampling solutions for the challenging problem of unbalanced training class distributions in training sets. This thesis is divided into nine chapters and based on the sub-problems considered in problem domain,the main contributions fall under three broad categories:

The three broad categories are Informative Sampling-oversampling (Chapter 4), Informative sampling-undersampling (Chapter 5 and Chapter 6) and Preprocessing +Informative-oversampling (Chapter 7 and Chapter 8).

Chapter 1: *Introduction*

This chapter provides actual problem definition followed by the classification algorithms used in the thesis for validating proposed resampling solutions.

Chapter 2: *Literature Review*

This chapter describes the taxonomy of the solutions proposed in the literature and recent developments and the research works carried out in the field of unbalanced data classification problem.

Chapter 3: *Performance Measures and Synthetic Datasets*

This chapter presents the performance measures used for evaluating the clas-

sifier performance on proposed methods and the synthetic data sets generated for validating the viability of proposed methods.

Chapter 4: *Extreme Outlier Elimination using Hybrid Sampling Technique*

This chapter provides the motivation and design of new hybrid oversampling method named Extreme outlier elimination using Hybrid sampling technique. A new concept named extreme outliers in minority class was introduced. We identified and eliminated the extreme outliers and then hybrid of SMOTE + RUS was applied. The validation of the proposed hybrid approach is tested with Insurance Fraud Detection dataset on RBF network and NB and on local classifier namely kNN. The efficiency of classifier was shown using TP_{rate}, TN_{rate} and classifier $G - mean$.

Chapter 5: *Majority Filter-based Minority Prediction (MFMP)*

This chapter presents the design of a new passive sample selection method named Majority Filter based Minority Prediction (MFMP) for selecting majority class informative instances for final classification. Proposed approach is demonstrated by conducting experiments on UCI repository datasets and compared with random undersampling. The efficiency of classifier was shown using minority class $recall$, $precision$ and $F - measure$.

Chapter 6: *Probabilistic Active Learning Approach*

This chapter discusses a new probabilistic active learning algorithm named CStatQSVM to counter the problem of loss of informative instances in Support Vector Machine classifier. Proposed approach is compared with LOB active learning method and other conventional methods namely random undersampling, Different Error Cost (DEC) [126] that solve class imbalance problem. Comparison across different methods is carried out with Friedman's ranking as well as Wilcoxon signed rank test.

Chapter 7: *Is PCA Effective for Preprocessing Unbalanced Data?*

This chapter illustrates the empirical study that is carried out to investigate whether PCA is effective for preprocessing the unbalanced datasets. This study initially derived conclusions on simulated datasets and further validated on 10 benchmark datasets from UCI repository.

Chapter 8: *CPC_SMOTE: A Class-Specific Dimensionality Reduction Framework*

From the conclusions of chapter 7, this chapter discusses a class-specific principal component framework for alleviating the bias caused by majority class variance in PCA subspace. Proposed framework is validated on 6 UCI and UCD repository datasets and compared with classical PCA and SMOTE preprocessing techniques.

Chapter 9: *Conclusion and Future Work*

We summarize the major contributions of the research work carried out in this thesis. We also highlight the future scope and further research directions.

This thesis covers both the theoretical aspects as well as applied aspects of data Mining and Machine learning research carried out in unbalanced class distribution problem. Theoretical characterization of the proposed resampling methods has been done through algorithmic representation. The applied aspect of the research reported in the thesis is reflected in the extensive experimentation conducted on the benchmark datasets related to Insurance Fraud Detection, UCI and UCD repositories. The thesis touches on both traditional as well as soft computing techniques.

Chapter 2

Related Work

In this chapter, we review relevant literature on how to handle unbalanced training set distributions for the classification problem. As some datasets that are available in real world have unbalanced classes, solving this class imbalance problem is critical in data mining community. Three workshops were organized at conferences on the class imbalance problem: AAAI- 2000 workshop [60], ICML-2003 workshop on Learning from Imbalanced Data Sets (II) [21], PAKDD-2009 workshop on Data Mining [23]. Furthermore, a special issue was published [22] to encourage research in unbalanced data classification. Two elaborate surveys [56, 111] and several short reviews [51, 121] were written to explore the current status of the research carried out in class imbalance problem. This chapter mainly deals with the effect of unbalanced datasets on the performance of various classifiers (learning difficulties) and the solutions till now proposed to address class imbalance problem. This chapter presents two different perspectives of the class imbalance problem: 1) identifying learning difficulties in classification algorithms due to unbalanced training set distributions (presented in section 2.1) and 2) new solutions for countering unbalanced training set distributions (elaborated in section 2.2).

2.1 Effect of Unbalanced Datasets on Classifier Performance

Researcher empirically analyzed the difficulties of several classification algorithms with unbalanced datasets. The empirical studies were carried out on synthetically simulated datasets with different data characteristics such as overlapping

classes, size and concept complexity as well as on real world datasets with different imbalance ratios. Most of the researchers concluded that the hardness of the class imbalance problem is not only due to class discrimination property of the classification algorithm being used, but also due to the internal data characteristics. Japkowicz et al. [63, 65] have reported that the hardness of the class imbalance problem is relative with respect to the concept complexity, imbalance ratio, size of the training set, minority class disjuncts and classification algorithm used for learning. Their study also has indicated that the Support Vector Machine (SVM) is less sensitive to unbalanced distributions compared to decision tree and neural network classifiers. However, Wu et al. [139] empirically have proved for highly unbalanced datasets that the SVM boundary is skewed towards positive class with less support vectors. From the studies of Batista et. al. and Jo et al. [5, 65, 91, 92], the performance of the decision tree classifier depends on the minority class degree of overlapping with respect to majority class distribution and minority class small disjuncts.

Consequently, Anand et al. [3] have observed that the class imbalance leads to slower convergence rate in Back Propagation NN and there is significant increase in minority class error rate whereas decrease in majority class error rate. Theoretically, the authors have justified that decrease in majority class error rate only is due to the downhill direction of its gradient vector.

Research of Fernandez et al. [41] have shown that Fuzzy Rule Based Classifier System ($FRBCS$) improves the classification performance over balanced datasets compared to original unbalanced datasets. Adding to this conclusion, the study has revealed that for highly unbalanced datasets $FRBCS$ is less sensitive to unbalanced distributions compared to DT algorithm. In another contribution, Garcia et al. [47] have pointed out that (kNN) classifier is sensitive to global imbalance ratio and concept complexity of the training set, when compared to local imbalance ratio. Their study also has reported that on local imbalance ratios kNN classifier better identifies the minority class compared to global methods such as Multi layer perceptron, NB and DT classification algorithms.

The application of Linear Discriminant Analysis (LDA) is studied for the class imbalance problem as well. Xie and Qie [141] have proved theoretically and empirically that LDA with the assumption of Gaussian distribution biases towards the majority class due to unequal covariance matrices. Further, the authors suggested

that balancing the class distributions can alleviate this problem. However, Hao and Titterington [52] have disproved the earlier claim on LDA that unequal size covariance matrices is a key reason for performance degradation in LDA. Contradicting the results of Xie and Qie, Hao and Titterington have empirically proved that balancing the original unbalanced distribution causes negative effect on LDA.

2.2 Methods for Handling Unbalanced Datasets

Several researchers worked on unbalanced data problem using standard machine learning algorithms such as Neural networks, Decision Tree and Support vector Machine. It is commonly agreed that these algorithms cause heavy bias towards the majority class. Among these algorithms, some are less sensitive and some are more sensitive towards the imbalance nature [63]. Solutions are proposed to counter the problem at data level and algorithmic level. Hybrids of both levels also exist in the literature. Based on the research solutions devised so far to address the problem of training unbalanced datasets we the extended the taxonomy shown in Fig 2.1 that is presented in [111].

2.2.1 Data Level Solutions

At data level, the performance deterioration due to unbalanced class is countered by changing the training set distribution using resampling techniques. Most of the research in class imbalance problem is centered on developing efficient resampling solutions, because they can be applicable for any classifier. Adoptable resampling solutions for unbalanced training sets are of two kinds. They are
Undersampling:-Balancing the class distribution by ignoring few majority class samples.
Oversampling:- Balancing the class distribution by increasing the number of minority class samples.

Undersampling

Undersampling is quite a popular technique to counter class imbalance problem. Random and informative are two types of undersampling techniques proposed in the literature.

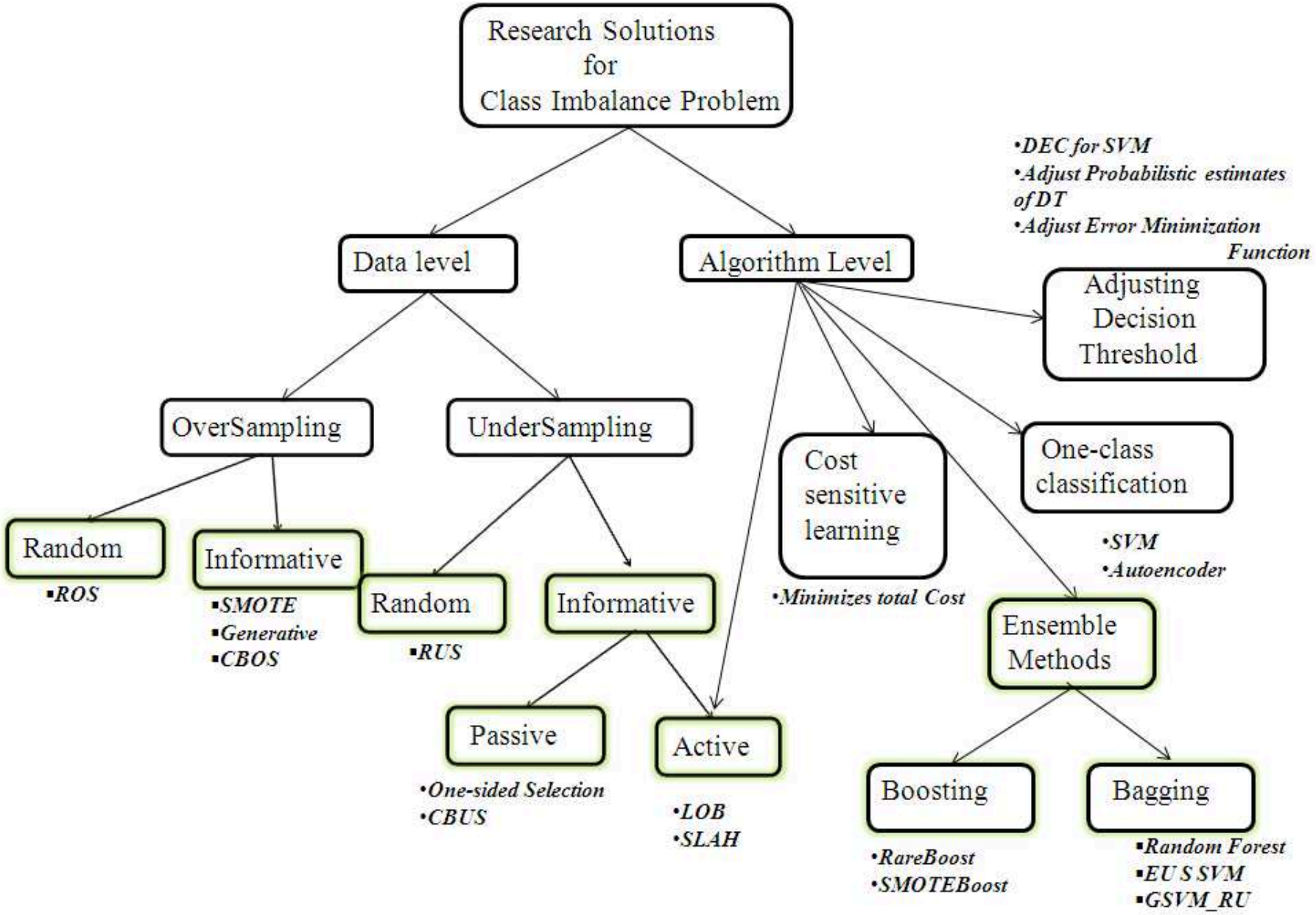


Figure 2.1: Taxonomy of research solutions to address class imbalance problem. (Abbreviations of various algorithms are mentioned in the text)

Random undersampling (RUS): Randomly selects equal number of minority class and majority class samples to make balanced class distributions.
Informative Undersampling: This method selects only the required majority class instances based on a pre-specified selection criterion to make balanced class distributions.

Informative sampling can be passive or active. Passive selection methods are proposed as preprocessing technique for selecting informative samples for a classification model whereas in case of active selection methods, informative samples are queried during the construction process of classification model.

Focusing on passive selection methods, Kubat and Matwin [70] have proposed *one-sided* selection approach for informatively sampling the majority class data. Based on the local proximity they have categorized the majority class into noise, borderline, redundant and safe samples. Later they adopted Hart's CNN (Condensed Nearest Neighbour) [55] rule for identifying safe and redundant majority points in the dataset as well as used the Tomek link [117] concept for eliminating noise and border points.

Zhang and Mani [150] have discussed 3 near-miss methods and one distinct method for selecting majority samples from the unbalanced class distribution. The near-miss methods select the majority points from those that are close to all or some of the minority points. The distinct method selects the majority points that are farther from all minority points. Yen and Lee [144] have described a cluster-based undersampling (CBUS) technique for unbalanced class distribution. The authors adopted random undersampling technique and the techniques proposed in [150] are on individual clusters rather than on whole data. In this approach imbalance ratio is used as a guide to select fixed number of points from each cluster to balance the training set distribution.

Cohen et al. [29] have proposed prototype selection based undersampling to reduce the majority class data. In order to produce balanced class distribution they partitioned the majority data into n number of clusters, where n is the number of minority samples in the data. Each of these n number of majority clusters are replaced with corresponding cluster centers such that the distribution is balanced. Yuan et al. [146] have proposed support cluster machine (SCM) algorithm based on kernel k-means clustering prototype selection method to handle large scale unbalanced datasets. In this approach the prototype selection is performed

in feature space rather than input space. The prototypes that are near to the hyperplane are retained for final training and the prototypes that are far from the hyperplane are rejected based on a threshold. Several ensemble methods exist in the literature for selecting informative instances. It can be observed that from the systematic studies of Ertekin et al. [39] active learning, Learning on the Border (LOB) yields balanced class distribution in the early rounds without loss of informative instances from both classes. Doucette and Heywoody [33] have proposed Simple Active Learning Heuristic (SALH) consists two components, subsampling and robust fitness function design for the case of GeneticProgram (GP) classification. The subsampling with uniform probability under a class balance enforcing rule for fitness evaluation is carried out with Dynamic Subset Selection [48] (DSS) model. Further, proposed approach is evaluated on Wilcoxon-Mann-Whitney (WMW) statistic which is form of AUC performance metric.

Oversampling

Oversampling is a frequently used technique to solve lack of information problem in learning caused by minority class. Random and informative are the two types of oversampling techniques proposed in the literature.

Random Oversampling (ROS): Replicating minority class samples is performed to make balanced class distribution.
Informative Oversampling: This method synthetically generates minority class instances based on a pre-specified criterion.

Random oversampling can directly applied for solving unbalanced data classification problem [16] but leads to overfitting [5]. Chawla et al. [18] have introduced a novel oversampling technique named Synthetic Minority Oversampling Technique (SMOTE). SMOTE synthetically generates minority samples by interpolation across line segment of the k minority class nearest neighbours, considering one minority sample at a time. Depending upon the amount of oversampling required, neighbours from the k nearest neighbours are randomly chosen. The main aim of SMOTE is to effectively force the decision region of the minority class to become more general. Further, the authors proposed a hybrid of SMOTE and Random Undersampling. SMOTE alone and hybrid of SMOTE + RUS were successfully applied on several real world applications like binding site prediction

[109] in bioinformatics, Drug Event Predictive Models in Labor and Delivery [113], Intrusion detection [27]. Their experimental results have shown that the combined technique of SMOTE and Random UnderSampling on DT obtained better generalization over all other classifiers and exhibits better classifier performance over random undersampling.

Several variants of SMOTE were also discussed in the literature. Han et al. [54] devised two different extensions of SMOTE named Borderline $SMOTE\_1$ and Borderline $SMOTE\_2$. SMOTE blindly generates synthetic sample around all minority samples, where as borderline-SMOTE selectively generates synthetic samples across the boundary points between the minority and majority regions. To identify the border points they divide the minority points into *noise*, *safe* and *boundary points*. A *safe point* is that point whose minority class nearest neighbours are more than majority class nearest neighbours. A *boundary point* is surrounded by majority class nearest neighbours than minority class nearest neighbours. Borderline $SMOTE\_1$ uses only minority border points for SMOTEing where as Borderline $SMOTE\_2$ uses both majority and minority border points. Enhanced versions of SMOTE [128, 148] for lower dimension [128] as well as higher dimension [148] have also been proposed.

Batista et al. [5] have designed new hybrid sampling solutions using SMOTE and data cleaning methods like Condensed Nearest Neighbour rule [55], Edited Nearest Neighbour rule [135] and Tomek links [117] for small unbalanced training sets. The designed hybrids are CNN + Tomek links, SMOTE + Tomek Links, and SMOTE + ENN methods. Classifier performance and syntactic analysis using mean number of rules as well as mean number of conditions per rule on pruned decision tree classifier outperformed the un-pruned decision tree over devised hybrids. Liu et al. [77] have proposed a *Generative Oversampling* method, where the synthetic samples are generated by learning minority class data on probabilistic models. Proposed method outperformed popular SMOTE algorithm on text mining task. Nickerson et al. [90] have developed a new guided sampling approach to address within the class imbalance problem named cluster based oversampling (CBOS). In this approach clustering technique guided to identify the subcomponents in both classes. Later the algorithm inflates all clusters excluding the largest cluster, with the size of the largest cluster so that training set distribution is balanced. For detailed explanation consider there are 4 and 3 clusters from majority

and minority classes respectively and,
No. of samples in each majority clusters: 5 5 5 15
No. of samples in each minority clusters: 2 4 5

After cluster based oversampling,
No. of samples in each majority clusters: 15 15 15 15
No. of samples in each minority clusters: 20 20 20

Here the majority class clusters that are not largest in number of samples, are inflated with the size of the largest majority class cluster whereas each minority class clusters is inflated with $\frac{no.of majority class samples in all clusters}{no.of minority clusters}$, here 60/3 =20 samples.

They conducted experiments on text classification domain and their results show that, the proposed approach works well for within-the class imbalance domains.

Researchers have conflicting views about the effectiveness of performance of Oversampling and Undersampling techniques.

Japkowicz and Stephan [63] as well as Batista et al. [5] have reported that over sampling is better than undersampling. Furthermore, Ling and Li [73] as well as Drummond and Holte [35] have reported that undersampling yields superior performance. Ling and Li [73]have explored random undersampling and hybrid of random undersampling and oversampling for direct marketing problem. They reported that best lift index is obtained by undersampling the majority class only [1]. Drummond and Holte [35] evaluated undersampling and oversampling using cost curves and indicated that undersampling beats oversampling. Consequently a study from [131] has indicated that the naturally occurring distributions (unbalanced) are not best for learning. According to their experiments, the optimal distribution for learning lies between 50% and 90% of minority class samples in the training set. They suggested that maximum area under ROC curve (AUC) can be attained by balancing unbalanced training set distributions. However, this leads the classifier to attain AUC better than natural distribution but not optimal one. However, study of "whether oversampling is more effective than undersam-

[1]Lift charts are widely used performance measure in market analysis problems. Lift charts show the dependency between cost and expected benefits over target models in marketing applications.

pling "and "which oversampling or undersampling rate should be used"was done in [40], and finally suggested that combining different expressions of the resampling can leads to optimal distribution for learning. They have proposed a three-level architecture which consists of (see Fig 2.2)

Classifier level: The elimination of classifiers takes place from pool of classifiers based on a weighing scheme.

Expert level: A combination scheme is applied on each expert which constitute either pools of oversampling classifiers or undersampling classifiers.

Output level: A combination scheme is applied on the result of oversampling expert and undersampling expert.

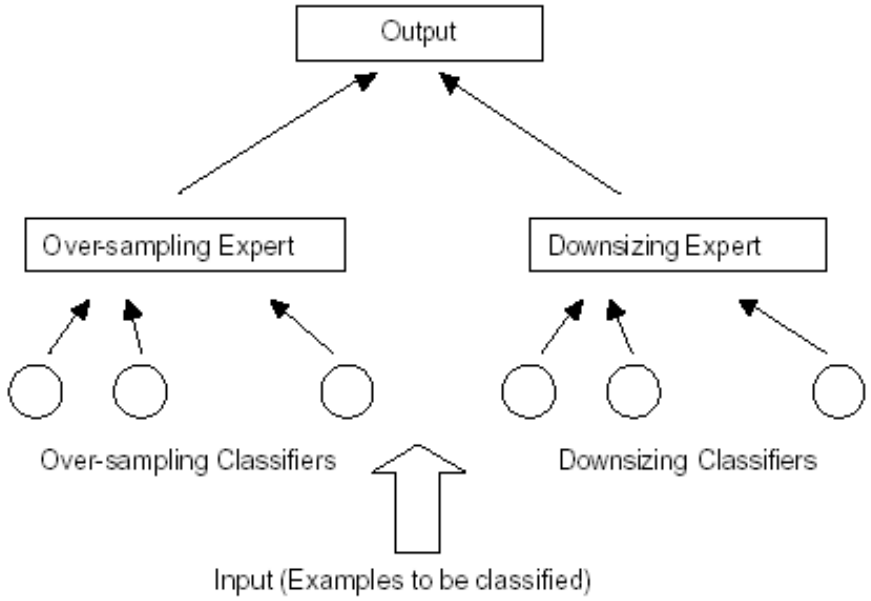


Figure 2.2: Three level architecture for multiple resampling (taken from [40])

This combination system shown in Fig 2.2 works as follows, If one of the non-eliminated classifiers either from oversampled expert or undersampled expert decides that an example is positive, the overlying expert also decides the same. Similarly, if either one of the two experts decides (based on its classifiers' decision) that an example is positive, so does the output level, and thus, the example is classified as positive by the overall system. Recently, Chawla et al. [24, 25]

have introduced a wrapper infrastructure in order to identify optimal sampling technique and parameters for a given data set. Proposed approach that applies cross-validation to first identify optimal undersampling percentage. Then with the undersampling percentage fixed, the wrapper approach finds the SMOTE percentage. This approach is based on the authors previous work [18]

From the above three studies, random oversampling and random undersampling, are widely used as a solution for addressing the class imbalance problem. But some erroneous situations may arise by using these methods. Some times random oversampling leads to over-fitting in the classification process [5] and random undersampling leads to information loss from the majority class [1, 35].

2.2.2 Algorithm Level Solutions

The algorithm level solutions adapt existing learning algorithms by giving more emphasis to small class while learning. The algorithm level solutions include (a) Novelty detection (One_class classification) methods where a boundary is drawn around target concept (b) adjusting the decision threshold of the classifier in order to give higher priority to the minority class in objective function and (c) cost sensitive learning where higher misclassification costs are assigned to minority class.

One_class classification

The contributions in [62, 76, 99] have explored novelty detection methods for unbalanced training set problem and concluded that for highly unbalanced datasets novelty detection approaches are effective than discriminative learning methods. The novelty detection methods concern with one-class classification of target class by constructing a boundary around it and try to discriminate outliers from target class. For unbalanced training sets, target class constitutes the minority class where as outliers correspond to majority class, ofcourse the opposite formulation is also possible. Several novelty detection methods like one-class Support Vector Machine ($OCSVM$) [103], Support Vector Data Description ($SVDD$)[114], Linear Vector Quantizer- Novelty Detection ($LVQ-ND$) [75] were devised by machine learning researchers. However, $OCSVM$ and $SVDD$ were successfully applied to the applications of class imbalance problems [76, 99] which are highly unbalanced in nature. The success of novelty detection methods on unbalanced

training sets purely depends on boundary threshold between target class and outliers. Too strict threshold can aggravates minority class prediction problem by leaving the minority samples outside the boundary whereas too loose threshold can aggravates majority class prediction by incorporating the majority samples inside the boundary. Wu and Ye [140] have modified $SVDD$ by constructing the small enough sphere around target class and maximizing the boundary between target samples and outliers as large as possible.

Ensemble based Techniques

Popular ensemble methods namely bagging [7] and boosting [102] and Stacking [138] were explored to boost minority class predictive accuracy. Ensemble methods try to improve classifier performance than single classifier by generating diversity around classification models.

Bagging concurrently learns individual classification models of same classifier from bootstraps of training sets. Several variants of bagging were proposed to counter loss of data problems from majority class. Chen et al. [14] have adopted resampling methods on Random forests [8] to boost the performance over unbalanced datasets.

Kang and Cho [72] have discussed an approach named Ensemble of undersampled SVMs (EUS SVM), an SVM ensemble on the samples generated by random undersampling. Tang and Zhang [112] introduced the idea of Granular Support Machine- Repetitive undersampling (GSVM_RU), which repetitively learns SVM model by adding majority class SV's to the minority class until there is no improvement in the classification performance. The majority class SV granules are obtained from a set of SVM models. To these set of SVM models training set for $(n+1)^{st}$ SVM model is one of the random undersamples combined with majority class SV's from n^{th} SVM model. A new SVM model is added to the set until there is not much performance gain.

Yoon and Kwek [145] have proposed a recursive clustering technique for partitioning the majority class data into pure majority clusters for functional genomics problems. Later the pure majority clusters are pruned and each impure minority cluster is exposed to an individual classification model of NN meta-model. Altinay and Ergun [2] have initially balanced the class distribution using the prototype selection method [29]. Furthermore, they modified the weights of the Adaboost

algorithm according to the average samples in each majority cluster for speaker verification. Chan et al.[13] have used stacking strategy to combine the predictions from multiple classifiers namely DT, $CART$, *Ripper* and NB drawn on smaller subsets of the training sets to improve the cost saving for credit card fraud detection. Phua et al. [95] further refined the model has proposed in [13] by a stacking-bagging approach to improve insurance fraud detection cost saving. Stacking combines the predictions from different learning algorithms based on the metadata formed by the predictions of those different algorithms.

Boosting iteratively boosts a weak learner by emphasizing on misclassified training instances of previous models in terms of different weights over each run. Actually this weighting strategy of samples in AdaBoost is equivalent to resampling the data space that includes both up-sampling and down-sampling. Algorithm 1 describes the stepwise approach for Adaboost algorithm. Initially all samples are equally weighted with $1/N$ where N is the number of samples in training set. For M number of iterations the weights of the samples are updated as shown in eq. 2.1 and 2.2. The final classification decision is made as shown in eq. 2.3.

Despite the fact that AdaBoost treats classified and misclassified samples equally, Joshi et al. [67] have introduced separate weight update rule for both positive and negative examples (RareBoost) so that the classifiers' *precision* and *recall* are treated equally. The main intuition behind this criterion is that learning better models to distinguish, False Negatives (FN) from True Negatives (TN) leads to good *recall* where as distinguishing False positives (FP) from True Positives (TP) leads to good *precision* for a classifier. They also enhanced the SLIPPER algorithm, which abstains from making any decision on some training examples by either to choose positive or negative class examples dynamically. Thus the research of Joshi and his colleagues focus on improving *precision* and *recall*. Researchers modified AdaBoost to support cost sensitivity by incorporating cost functions inside the weight updating step 2.2 in Algorithm 1. There are three ways to introduce cost function in 2.2 inside the exponent, outside the exponent and both inside and outside the exponent. Fan et al. [42] for the first time incorporated cost function in r_t and inside the exponent of eq. 2.2 to accommodate the cost sensitivity, thus increasing the weighted cost for misclassified samples and decreasing the weighted cost for correctly classified samples in successive boosting iterations.

Algorithm 1 Psuedocode for AdaBoost algorithm (Taken from [67])

Input:Training set $T = \{(x_i, y_i)\},\ where\ i = 1....N$;

$x_i \epsilon X$, $y_i \epsilon -1, 1$;

/* $X = set\ of\ attributes, y_i = corresponding\ class\ label * \backslash$

Number of iterations=M, Weights $D_1(i) = 1/N$;

for t=1....M **do**

Learn a weak model h_t using D_t;

Compute Weight $\alpha_t = \frac{1}{2} ln(\frac{1+\tau_t}{1-\tau_t})$; where

$$\tau_t = \sum_{i=1}^{N} D_t(i) h_t(x_i) y_i; \tag{2.1}$$

Update Weights

$$D_t(i+1) = (D_t(i) exp(-\alpha_t y_t h_t(x_i)))/Z_t; \tag{2.2}$$

where Z_t is chosen such that $D_t(i+1) = 1$

end for

Final Model:

$$H(x) = sign(\sum_{i=1}^{M} \alpha_t h_t(x)); \tag{2.3}$$

In another contribution, Ting [119] has incorporated cost in 3 different expressions of eq. 2.2 and named the variants as CSB0, CSB1, CSB2. Furthermore, Sun et al. [110] have enhanced the cost sensitive boosting by devising three cost sensitive variants AdaC1, AdaC2 and AdaC3 based on the three ways to incorporate cost (in eq. 2.2) with appropriate modifications. A very good comparison between cost sensitive boosting algorithms is also provided in [110].

Chawla et al. [20], have launched sampling technique on boosting procedure, by generating synthetic samples using SMOTE procedure around misclassified minority class examples in each run of boosting in SMOTEBoost. This forces the boosting algorithm to concentrate more on the minority class samples, which results in better minority class prediction with out disturbing the majority class distribution. Guo and Viktor [50], have proposed another variant of sampling based boosting, by generating synthetic samples for both majority and minority classes. For each run of boosting procedure, this algorithm identifies the hard examples for learning from both minority and majority classes. Among these hard examples few samples are picked as seed points as shown in equations 2.4 and 2.5.

$$Maj_s = \min(Trainingset_{IR}, No.of\ hard\ samples\ from\ majority\ class) \quad (2.4)$$

$$Min_s = \min(Trainingset_{IR} * Maj_s, No.of\ hard\ samples\ from\ minority\ class) \quad (2.5)$$

In eq. 2.4 and eq. 2.5, IR represents imbalance ratio $\frac{majority\ class\ samples}{minority\ class\ samples}$ of the training set and Maj_s as well as Min_s represent seed points. Here IR guides in selecting seed points in every boosting run. Finally, $Maj_s * majority\ class\ samples$ and $Min_s * minority\ class\ samples$ number of synthetic samples are added to training set and the sample weights are updated according to seed point weights in every round of boosting. However, adding new points in every run leads to extra computational overhead in this approach. Liu et al. [79] have introduced two ensemble methods on the samples generated by random undersampling. One ensemble method, EasyEnsemble [79], learns an AdaBoost ensemble over each sample and finally combines their outputs. Another method, BalanceCascade [79] is also similar to EsayEnsemble except that it removes the correctly classified instances in further rounds of the algorithm. Recently, Wang and Japkowicz [129] have introduced the idea of boosting Different Error Cost (DEC) of SVM hyperplane for further enhancing the SVM classifier performance towards unbalanced

datasets.

Cost Sensitive learning

As cost sensitive learning assigns different costs for misclassification errors [82], instead of altering the class distribution, cost sensitive learning is adopted to counter the unbalanced training set distributions. The misclassification costs associated with each type of error is depicted in terms of cost matrix (Table 2.1).

Table 2.1: Cost Matrix

	PredictedNegative	PredictedPositive
Actual Negative	C_{TN}	C_{FP}
Actual Positive	C_{FN}	C_{TP}

In case of unbalanced distributions the misclassification costs are unequal, i.e the misclassification cost associated with minority class is greater than the misclassification cost associated with majority class ($C_{FN} > C_{FP}$) in order to improve minority class prediction rate. In contrast, the cost associated with correct classification is set to $C_{TN} = C_{TP} = 0$. Usually the cost items are unknown *a prior*, the goal of cost sensitive learning is to *minimize the total misclassification* cost for varied cost ratios of $C_{FN} : C_{FP}$ over cost matrix. The cost items can be incorporated either at data level or at algorithms level of classification process. At data level the cost sensitivity is achieved by weighing the data space with corresponding misclassification cost. Based on translation theorem [147], the total cost for misclassification in cost space is equivalent to total misclassification error in data space. Limited number of samples were selected through bootstrap sampling criterion so that total misclassification cost can be minimized.

Cost items can also be incorporated at the time of learning phase Ling et al. [74] introduced cost sensitivity in decision tree splitting criteria, rather than minimizing entropy. Here total misclassification cost is minimized at each split. Another work reported that tree can be pruned to minimize the misclassification costs [4]. Further, Drummond and Holte [34], have shown that the commonly used decision tree splitting criteria using the impurity measures such as accuracy, entropy, Gini index and DKM (Dietterich, Kearns and Mansoor) measure are relatively insensitive to cost and among the splitting criteria, DKM is cost insensitive whereas accuracy is inherently cost sensitive.

Similarly the cost can be introduced at four stages of neural network architecture [71], first is at probabilistic estimate; second is at neural network outputs; in third one cost items are incorporated in learning rate η; and at fourth one the error minimization function is modified for minimizing total misclassification costs. The effect of oversampling, undersampling, threshold-moving, hard-ensemble, soft-ensemble, and SMOTE in training cost-sensitive neural networks are studied empirically in [151]. Their study concluded that cost-sensitive learning is difficult for multi class classification and existing sampling approaches are efficient in increasing the efficiency of two-class classification. Finally, the cost items can be incorporated in meta-learning framework based on Bayes risk minimization to minimize the total misclassification cost [32].

Adjusting Decision Threshold

These methods adjust the bias of the classifier towards minority class by changing the objective function of the classifier. Morik et al. [87] and Veropoulos et al. [126] have proposed two different error costs (DEC) C^+ and C^- for SVM classifier in place of the usual single error cost. Morik et al. designed DEC on L1 norm whereas Veropoulos et al. on L2 norm. Recently Yang at al. [143] extended Veropoulos et al. model by incorporating margin compensations for error costs. They have replaced the error costs C^+ and C^- with new loss functions ξ_p^+ and ξ_n^-. The loss functions ξ_p^+ and ξ_n^- are equivalent to the paired constants (C^+, a^+) and (C^-, a^-), respectively. In these paired constants, C^+ and C^- are analogous to Veropoulos et al. error costs whereas as a^+ and a^- are newly incorporated margin compensations for the error costs. Liu et al. [80] have introduced weights, which indicates prior knowledge about each sample into Pawlak's [93, 94] rough set model. These weights are associated with each sample and further they proposed a weighted rough set model for classification. Xu et al. [142], have extended the Ishibuchi et al. [61] rule generation method named E-algorithm. E-algorithm normalizes the rule confidence and support measures with individual class percentage. E-algorithm was successfully applied on Duke Energy distribution outage data for cause identification.

Other Methods

Visa [120] has proposed a data driven fuzzy classifier based on mass assignment theory of the probabilistic fuzzy sets. They proved that the proposed classifier is more effective than the decision tree and neural network classifiers in handling imbalance datasets. They evaluated the fuzzy classifier on real world and synthetic datasets. The synthetic datasets constitute data from various levels of imbalance, concept complexity, overlapping percentage and dataset size. They reported that excluding high degree of overlapping and complete lack of data for minority class scenarios, proposed fuzzy classifier performed well for large imbalance datasets also.

2.2.3 Contributions from this thesis

This thesis proposes more focused sampling solutions for unbalanced data classification problem. As the real world datasets suffers from unbalanced and noisy data [115] this thesis proposes a new hybrid sampling technique of Extreme Outlier Elimination and Sampling Techniques as chapter 4 and the evaluation of the proposed method is presented with Insurance Fraud Dataset [95]. As undersampling suffers from information loss from majority class a more focused undersampling technique Majority Filter based Minority Prediction(MFMP) is presented in chapter 5. For Support Vector Machine *(SVM)* classifier as the oversampling causes extra computation overhead [39] and undersampling greatly affects the orientation of hyperplane [1], a Probabilistic Cost Weighted Active Learning approach is presented in chapter 6. A discussion on the viability of Principal Component Analysis (PCA) as a preprocessing technique for unbalanced datasets is presented in chapter 7. Based on the conclusions derived in chapter 7 a class-specific dimensionality reduction framework (CPC_SMOTE) is proposed in chapter 8. Finally conclusions and future research direction from this thesis is presented in chapter 9. Fig 2.3 depicts the roadmap of the contributions made by this thesis on the whole taxonomy of unbalanced datasets Fig. 2.1.

2.2.4 Chapter Summary

As the available real world data for classification task are of unbalanced classes, countering class imbalance problem becomes a crucial challenge in data mining

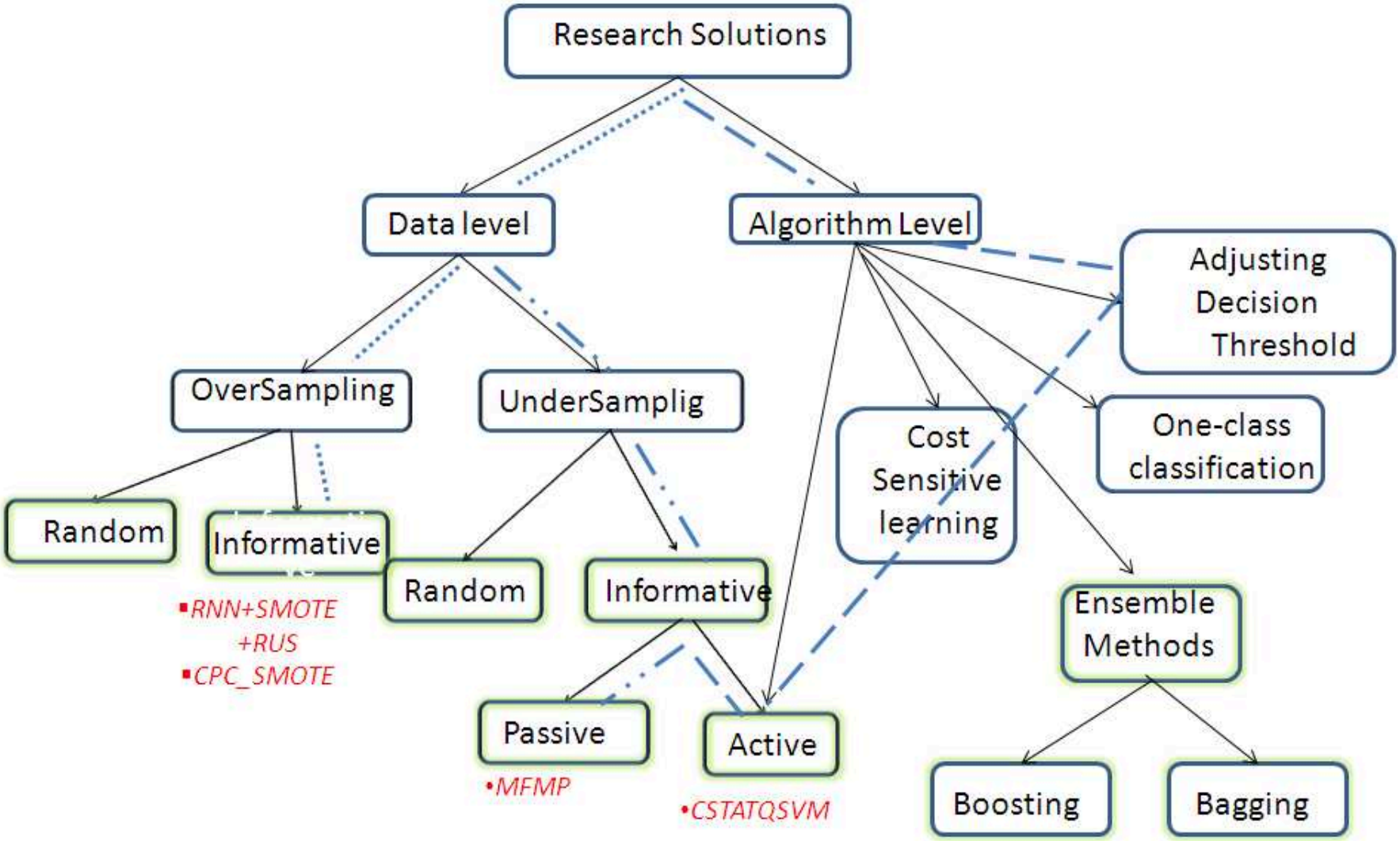


Figure 2.3: Roadmap of the proposed solutions in unbalanced data taxonamy

applications. In this chapter we have discussed the effect on classification model performance due to training with unbalanced data. Based on the learning difficulties faced by classification algorithms this chapter also discussed taxonomy of the research solutions to counter these unbalanced data classification problem. Although solutions were proposed at data and algorithm levels of the classification model learning process, resampling solutions are reported as prominent to alleviate the learning bias caused by unbalanced distributions. Ensemble based solutions also indicated performance improvement in classification models. The other solutions like cost sensitive learning and adjusting decision thresholds are yet to be explored for real world applications on a long scale.

In learning algorithm perspective decision tree and support vector machines were widely explored in scientific research as well as real world applications. Whereas the other classifiers like Linear Discriminant Analysis, Rough set based classifiers, Fuzzy classifier systems are yet to be explored in wider problems.

Chapter 3

Preliminaries

This chapter presents the overall background of the machine learning algorithms used for validating the proposed sampling methods. Further this chapter discusses the measures that are used for evaluating the classifier performance in case of two-class classification problem. Specifically, this chapter focuses on discussing the measures for unbalanced data classification problem and also presents the limitations of certain common measures when used with unbalanced datasets. Finally this chapter also presents the methods for comparing the multiple classifier performances across various datasets.

3.1 Machine Learning Classification Algorithms

Since this thesis proposes data level solutions for unbalanced data classification problem, several machine learning algorithms are used for validating the proposed sampling solutions. The classifiers we choose in this work are with varied characteristics such as local and global learning from a given dataset. The Decision Tree (DT), Naïve Bayes (NB), Radial Basis Function Networks (RBF) and Support Vector Machine (SVM) classifiers learn the data globally whereas k-Nearest Neighbour *(kNN)* classifier learns the data locally.

3.1.1 Decision tree (DT):

A decision tree [98] model is induced over training data for classifying unseen data. Decision tree is a tree like structure with root, non-leaf and leaf nodes. Each non-leaf node represents the condition or test on single value of an attribute;

the satisfactory of this condition leads to one or more sub-trees. Each leaf node labeled with one of the target variables for classification. A path from the root to leaf node represents a rule for classification. Decision tree is constructed in a top down fashion by using greedy technique. Initially the root node of tree points whole dataset. An information theoretic measure is applied at every non-leaf node for selecting the best attribute among all to split the dataset into subsets. The splitting of the non-leaf node process repeats recursively until the records in each subset belongs to singe target variable; label this subset with the corresponding target variable as a terminal node.

3.1.2 Naïve Bayes (NB):

Naïve Bayes classifier [36] is a simple probabilistic classifier based on Bayes theorem. It assumes conditional independence of all predictive variables with respect to target variable. Bayes theorem computes the product of prior probability and likelihood of occurrence of corresponding object i.e., posterior probability to classify unseen samples. The Naïve Bayes algorithm initially calculates the prior probabilities of all classes. The class prior probability distributions can be estimated with relative frequencies of each class from the whole training set distribution. The algorithm assigns new instance to a specific class, which attains maximum prior probability over all classes for that instance. Despite the conditional independence assumption, Naïve Bayes algorithm tends to learn quickly than other machine learning algorithms in many real world scenarios.

3.1.3 k-Nearest Neighbour (kNN):

kNN classifier [30] is an instance-based classification algorithm, which works on local learning. It classifies unseen instances based on the majority voting of the local instances from the training set. The term k in kNN classifier indicates the number of nearest neighbours to be considered for classifying the test instance. This classifier assigns a test instance, say z, to the highest voting class among z's k nearest neighbours. A similarity measure is applied to assess the closeness between the test and training instances in kNN.

3.1.4 Radial Basis Function Networks (RBF):

RBF [36] is one kind of neural network architecture constitute with three layers (i) input layer (ii) hidden layer (iii) output layer. Fig.3.1 presents the basic architecture of RBF neural network. All the input neurons are connected to hidden

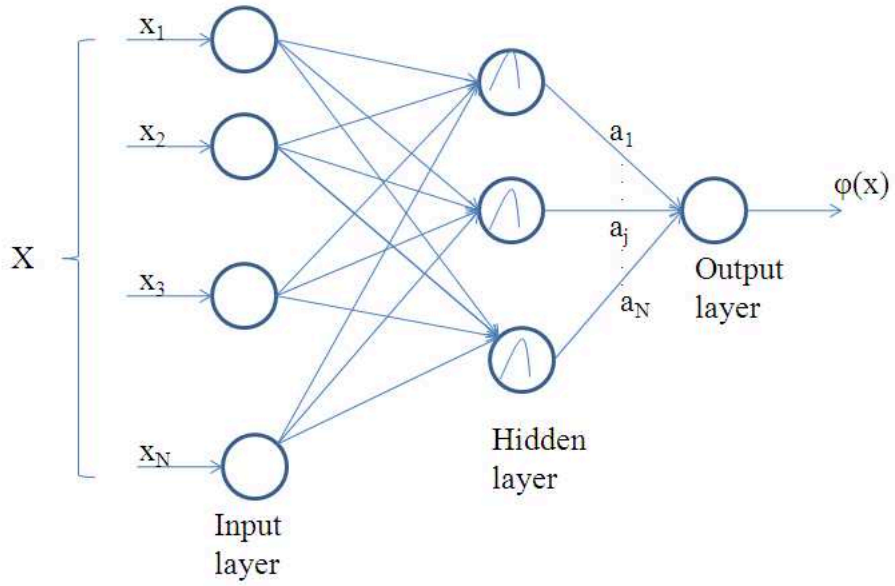


Figure 3.1: Architecture of Radial Basis Function Network

layer and neurons in hidden layer is connected to the output neurons. The linear output of the network $\varphi : R^n \rightarrow R$ is thus

$$\varphi(x) = \sum_{i=1}^{N} a_i \rho(\|x - C_i\|) \tag{3.1}$$

where N is the number of neurons in the hidden layer, C_i is the center vector for neuron i, and a_i are the weights of the linear output neuron. The non linear activation function of the hidden layer is the Gaussian function

$$\rho(\|x - C_i\|) = exp[-\beta\|x - C_i\|^2] \tag{3.2}$$

Here changing parameters of one neuron has only a small effect for input values that are far away from the center of that neuron. The weights a_i, C_i and β are determined in a manner that optimizes the fit between and the data.

3.1.5 Support Vector Machine (SVM):

Support Vector Machines [125] are very popular for their strong generalization capabilities and global optimization criterion. Fig. 3.2 depicts the training of an SVM classifier. Given a training set of N data points $\{(x_i, y_i)\}_{i=1}^{N}$ with input

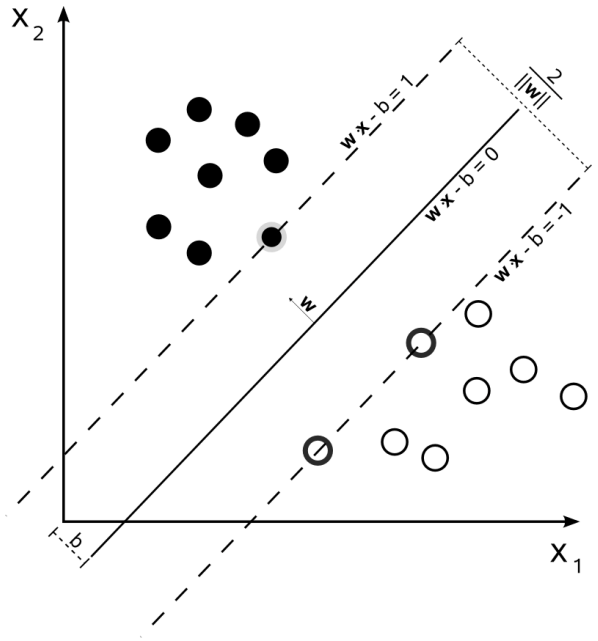


Figure 3.2: Training a Support Vector Machine,Where w is the norm of the hyperplane and the support are those patterns with the distance b from the hyperplane. The support vectors shown in circles. (Taken from [36])

data $x_i \epsilon R^N$ and corresponding binary class labels $y_i \epsilon \{+1, -1\}$, the SVM classifier describes an optimal hyperplane in feature space that separates the two classes by the largest margin. The objective function for minimization is given as

$$\min_{w,b,\xi_i} \frac{1}{2} w.w^T + C \sum_{i=1}^{N} \xi_i \tag{3.3}$$

$$subject\ to \begin{cases} \forall i\ y_i(w^T \Phi(x_i) + b) \geq 1 - \xi_i \\ \forall i\ \xi_i \geq 0 \end{cases} \tag{3.4}$$

Where w is the norm of the hyperplane, b is the intercept of hyperplane with origin, $\Phi(x_i)$ is the mapping of the input to feature space x_i, y_i is corresponding class label, ξ is the slack variable for handling nonlinearity and C is the loss function for misclassification cost. Here C is a tuning parameter. Generally, eq.3.3 and

6.2 are solved by convex Quadratic Programming Problem by formulating its dual cost function given as

$$\max w(\alpha) \equiv \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{3.5}$$

$$subject\ to \begin{cases} \forall i & 0 \leq \alpha_i \leq C \\ \forall i & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases} \tag{3.6}$$

Here α_i 's are the Lagrange multipliers whose values are non-zeros for the training instances that are at margin. Those training instances are known as support vectors. In eq.3, $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ represents the kernel matrix. After solving QP the norm of the hyperplane can be represented as

$$w = \sum_{i=1}^{N} \alpha_i \Phi(x_i) y_i \tag{3.7}$$

and the test instances are classified with

$$y(x) = sign[w.k(x_i, x) + b] \tag{3.8}$$

Equations eq. 3.7 and 3.8 show that support vectors play crucial role in defining SVM boundary.

3.2 Performance Measures

Usually, for two-class classification problem, the final possible outcomes of a classifier can fall into four categories and depicted in the form of a Confusion Matrix (also called a contingency table see Table 3.1). From Table 3.1 the four categories of classification outcomes are defined as follows, Table 2.1 presented in chapter 2 is similar.

Table 3.1: Confusion Matrix

	PredictedNegative	PredictedPositive
Actual Negative	True Negative(TN)	False Positive (FP)
Actual Positive	False Negative(FN)	True Positive (TP)

- True Positive rate TP_{rate} is the number of positive examples correctly classified.

$$TP_{rate} = Sensitivity = \frac{TP}{(TP + FN)}. \tag{3.9}$$

- True Negative rate TN_{rate} is the number of negative examples correctly classified
$$TN_{rate} = Specificity = \frac{TN}{(FP + TN)}. \tag{3.10}$$
- False Positive rate FP_{rate} is the number of negative examples incorrectly classified as positive.
$$FP_{rate} = \frac{FP}{(FP + TN)}. \tag{3.11}$$
- False Negative rate FN_{rate} is the number of positive examples incorrectly classified as negative.
$$FN_{rate} = \frac{FN}{(TP + FN)}. \tag{3.12}$$

TP_{rate} and TN_{rate} are also known as *sensitivity* and *specificity* respectively. From the above classification model outcomes, the measure for evaluating classifier performance is defined as

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)}. \tag{3.13}$$

$$Error\_rate = 1 - Accuracy. \tag{3.14}$$

For instance, consider a classification model that outputs TNrate=94% and TPrate=0% for an Insurance Fraud detection system where the data distribution contains 94% of non_fraudulent and 6% of fraudulent transactions. In this case the accuracy becomes 94% with sole contribution from majority class. But ignoring the fraud record predictions leads to a great lose to organizations. So obviously accuracy is not an appropriate measure for unbalanced datasets due to the biased nature of the classification performance towards majority class.

3.2.1 Performance Measures on Unbalanced Distributions

As accuracy is biased towards the majority class, class-specific measures and balanced performance measures are used for evaluating the classifier performance.

$F - measure$

The class-specific measures, *precision*, *recall* and $F - measure$ [100], are widely used for estimating the minority class prediction and they are defined as

$$recall = TP_{rate}. \tag{3.15}$$

$$precision = \frac{TP}{(TP + FP)}. \tag{3.16}$$

$$F - measure = \frac{2 * recall * precision}{(recall + precision)}. \tag{3.17}$$

Here *recall* is actually the true positive rate of how many number of positive (Minority class) samples are predicted correctly but it is not concerned with negative class (majority class) prediction. On the other hand, *precision* discusses how many number of samples in the whole distribution are predicted as positives (positive predictive value) which is a trade-off between TP_{rate} and FP_{rate}. Recall and precision goals are often conflicting [67] as the classification model is biased by TN_{rate} in unbalanced distributions. A harmonic mean of precision and recall named $F - measure$, depicts the trade-off between precision and recall. In case of $F - measure$, if both *precision* and *recall* are high, then $F - measure$ is also high, increasing *recall* rates without disturbing the *precision* of the minority class (target class) is a challenging issue. $F - measure$ is widely used throughout this thesis for evaluating the classifier performance.

$G - mean$

Unlike $F-measure$ which evaluates only single class performance at a time, Kubat and Matwin [70], suggested to use Geometrical mean of TP_{rate} and TN_{rate} named G-mean for evaluating entire model performance. As G-mean is the square root of TP_{rate} and TN_{rate}, it assesses balanced performance based on the assumption that True Positive Rate (TP_{rate}) and True Negative Rate (TN_{rate}) are supposed to be high simultaneously for yielding better classification models. G-mean was successfully applied for solving unbalanced data classification problem [1, 47, 69, 70]. The definition of G-mean is

$$G - mean = \sqrt{TP_{rate} * TN_{rate}} \tag{3.18}$$

Though, there are several measures that are available to evaluate the classifier performance in case of unbalanced data classification problem, this thesis uses minority class $F - Measure$ and whole classifier's $G - mean$. Minority class $F - Measure$ which reflects the minority class prediction which inherently indicates the trade-off between TP's and FP's. A high minority class $F-Measure$ indicates

that both minority as well as majority class predictions are high. Further $G-mean$ was also used in this thesis, on occasions where whole classifier performance was to be evaluated. Other performance measures are described in D.

3.3 Comparison of Performance of Classifiers

In order to provide correct empirical study across various datasets and on different methods, in this contribution we have made use of statistical techniques, specifically we have employed non-parametric tests. We describe the procedures for performing pair and multiple comparisons. Specifically, we have employed $Paired\ T-Test$ and Wilcoxon signed-rank test as a non-parametric statistical procedure for performing pairwise comparisons between two algorithms. For multiple comparison, we have used an Friedman test to detect statistical differences between different methods.

3.3.1 Paired T-Test

An usual way to test whether the average difference between two classifiers performance over varying datasets is significantly different from zero is $Paired\ T-Test$ [37]. The result of the $Paired\ T-Test$ indicates a rejection of the null hypothesis at the 5% significance level if the hypothesis value is one. The hypothesis of zero value indicates a failure to reject the null hypothesis at the 5% significance level.

Let $d_i = C_i^2 - C_i^1$ be the difference between the performance scores of two classifiers C_i^2 and C_i^1 on the i^{th} out of N data sets. The $Paired\ T-Test$ is computed as $\bar{d}/\sigma_{\bar{d}}$ and is distributed according to the student distribution with $N-1$ degrees of freedom. But $Paired\ T-Test$ suffers from the limitations like normality assumptions between the differences of two classifiers that are compared, commensurability and outliers.

3.3.2 Wilcoxon Signed-Ranks Test

Wilcoxon Signed-Ranks Test is an alternative to the $Paired\ T-Test$. It is a pairwise test [149] that aims to detect significant differences between the behavior of two algorithms. Let d_i be the difference between the performance scores of the two classifiers on i^{th} dataset out of N datasets. The differences are ranked

according to their absolute values; average ranks are assigned in case of ties. Let R^+ be the sum of ranks for the data sets on which the second algorithm outperformed the first, and R^- the sum of ranks for the opposite. Ranks of $d_i = 0$ are split evenly among the sums; if there is an odd number of them, one is ignored:

$$R^+ = \sum_{i|d_i>0} rank(d_i) + \frac{1}{2} \sum_{i|d_i=0} rank(d_i) \tag{3.19}$$

,

$$R^- = \sum_{i|d_i<0} rank(d_i) + \frac{1}{2} \sum_{i|d_i=0} rank(d_i). \tag{3.20}$$

Let T be the smallest of the sums, $T = min(R^+, R^-)$. If T is less than or equal to the value of the distribution of Wilcoxon for Nd_i degrees of freedom (Table B.12 in [149]), the null hypothesis of equality of means is rejected. The Wilcoxon signed ranks test is more sensible than the $Paired\ T - Test$ and less effective towards outliers.

3.3.3 Friedman's Test

The Friedman test [45], [46] is a non-parametric test that ranks the classification algorithms for each data set separately, the best performing algorithm getting the rank of n, the second best rank $n - 1$ and so on. In case of ties, average ranks are assigned. Here n is the count of maximum number of classifiers considered for comparison. Let r_i^j be the rank of the j^{th} of k algorithms on the i^{th} of N data sets. The Friedman test compares the average ranks of algorithms,

$$R_j = \frac{1}{N} \sum_i r_i^j \tag{3.21}$$

Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks R_j should be equal, the Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)} \Big[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \Big] \tag{3.22}$$

is distributed according to χ_F^2 with $k - 1$ degrees of freedom, when N and k are big enough (as a rule of a thumb, $N > 10$ and $k > 5$). For a smaller number of algorithms and data sets, exact critical values have to be computed.

The null hypothesis is rejected if the estimate χ_F^2 value is greater than the theoretical value, i.e., the algorithms are significantly different statistically.

3.4 Chapter Summary

This chapter discussed, the performance measures for evaluating the classifier performance in case of two-class classification problem, limitations in performance measures towards unbalanced data classification and finally the measures that are generally used for unbalanced data classification problem. Along with performance measures this chapter also discusses the background of machine learning algorithms used in this study and the methods for comparing the results of multiple classifiers across various datasets.

Informative Sampling - oversampling

Chapter 4

Extreme Outlier Elimination and Sampling Techniques

As described in section 2.2.1 of chapter 2 oversampling is one of the solutions to alliviate the bias caused by majority class. This chapter proposes an enhancement to a hybrid oversampling technique SMOTE+RUS. The remainder of this chapter is organized as follows. In Section 4.2 we discuss the background for Extreme outlier elimination and hybrid sampling. Section 4.3 describes the proposed approach and experimental setup. Section 4.4 provides the experimental case study with Insurance Fraud dataset [95], results and discussion. In section 4.5 we have summarized the chapter.

4.1 Introduction

Usually, classification algorithms exhibit poor performance while dealing with unbalanced datasets and results will be biased towards majority class. As described in 2.2.1, oversampling is one of the solutions to alleviate the bias caused by majority class. Usually oversampling balances the whole training set distribution by increasing the size of the minority class data. Oversampling is again of two types, 1) Random oversampling and 2) Informative oversampling. Random oversampling balances the training distribution by replicating random minority class samples whereas informative oversampling balances the training distribution by generating new synthetic minority class data based on heuristics. Several authors have reported that random oversampling [5, 18] leads to overfitting, as it replicates

exact copies of the minority class samples. Chawla et al. [18] have introduced a novel oversampling technique named Synthetic Minority Oversampling Technique (SMOTE). SMOTE synthetically generates minority samples by interpolation across line segment of the k minority class nearest neighbours, considering one minority sample at a time. Depending upon the amount of oversampling required, minority class neighbours are randomly chosen from k neighbours for new sample generation. The main aim of SMOTEing is to effectively force the decision region of the minority class to become more general. But if there are minority class samples that are far from actual minority regions, using these points for SMOTEing leads to mixing of classes. Therefore, degrading the classifier performance. Fig 4.1 depicts the synthetic sample generation criterion using SMOTE and specified limitation.

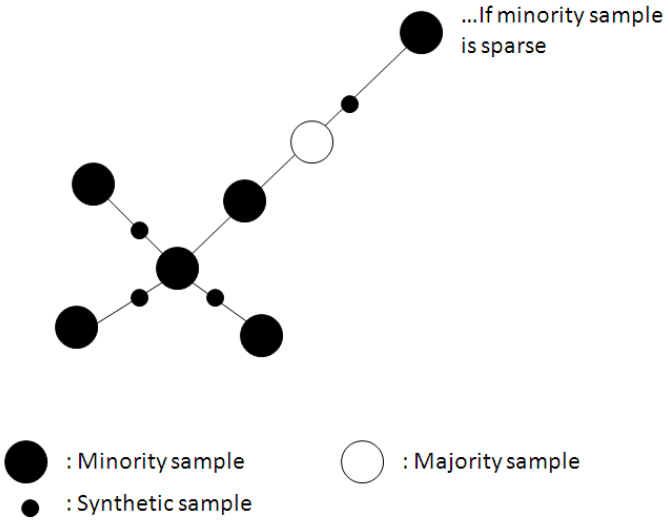


Figure 4.1: Synthatic sample generation in SMOTE

Mislabeled instances in training set has worsen the generalization capability of the predictive model [11, 88]. From [88] mislabeled instances which are different in its class label from its geometrical neighbourhood constitute special case of outliers. In order to improve classification accuracy, a filtering stage to identify and handle mislabeled examples followed by a prediction stage on the reduced training set is required As [11]. This forms the motivation for the proposed approaches in this chapter.

In this chapter we have discussed a hybrid sampling approach for unbalanced data classification problem. Proposed hybrid uses extreme outlier elimination us-

ing k Reverse Nearest Neighbours ($kRNN$) [104] approach with hybrid sampling technique, a combination of random undersampling of majority class samples and SMOTE to oversample the minority class samples. As $kRNN$ automatically captures the density around minority class point in minority class space, $kRNN$ based extreme outlier for minority class samples were defined. As the accuracy measure has bias towards majority class instances, the performance measures we could use here are $G - mean$, TP_{rate} and TN_{rate}. We will compare our approach with the traditional approaches, namely, random undersampling (RUS), random oversamplingn (ROS) and hybrid sampling that does not use any outlier elimination of minority samples.

4.2 Background

This section describes the background of the methods used for proposing extreme outlier elimination and Hybrid sampling approach.

4.2.1 Hybrid of Synthetic Minority Oversampling Technique and RUS

As described in 2.2.1 hybrid sampling of SMOTE and random undersampling are prominent solutions for unbalanced data classification problem [27, 109, 113]. SMOTE+RUS was employed to alleviate from the bias caused by majority class samples. Synthetic Minority Oversampling technique (SMOTE) was introduced by Chawla et al, [18] in which the minority class samples are over-sampled by creating synthetic (or artificial) samples rather than replicating random minority class sample. The following psudocode in Algorithm 2 describes the SMOTE algorithm. Latter combined approach of SMOTE and random undersampling (RUS) was devised by the authors of SMOTE to further improve the performance of the classifier towards unbalanced distributions. Sience SMOTE projects new samples between minority class geometrical nearest neighbours, minority class outliers (mislabeled samples) play a major role and some times the newly generated synthetic samples around actual minority class outliers leads to class mix. Thus hampers the classifiers generalization ability.

Algorithm 2 Pseudocode for SMOTE Algorithm (taken from [18])

Input:

N=Set of minority class samples; T=Smote factor;

k=Number of nearest neighbours;

Output=(T/100) * N synthetic samples; /* If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd $*\backslash$

Begin

if $T < 100$ **then**

Randomize the N minority class samples;

N= (T/100) * N;

T=100;

end if

T = (int)(T/100); /*The amount of SMOTE is assumed to be in integral multiples of 100 $*\backslash$;

for $i = 1$ *to* N **do**

Compute k nearest neighbours for i, and save the indices in the $nnarray$;

Populate(T, i, $nnarray$);

end for

Populate(T, i, $nnarray$) /* Function to generate the synthetic samples $*\backslash$

while $T \neq 0$ **do**

Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i;

compute the difference dif between i and nn;

Generate a random number gap between 0 and 1;

compute new synthetic sample as $synth = i + dif * gap$;

T=T-1;

end while

return; /* End of Populate $*\backslash$

End

4.2.2 Outlier Detection and Filtering by RNN

Several outlier detection and filtering methods are devised to filter mislabeled training instances for classification problem. Generally they are distance based [88] or classification algorithm based [11]. In this work k- reverse nearest neigbbour ($kRNN$) [104] based outlier detection and elimination was employed for SMOTE algorithm. The advantage of $kRNN$ over other distance based outlier methods is the parameter independency. In case of $kRNN$ the neighbourhood density around a sample p increases with the increase in number of neighbours k value.

Following are the notations used in this chapter.

Table 4.1: Notations and definitions used in this chapter

Notations	Definition
X	d-dimensional dataset
$kNN(p)$	Set of k-nearest neighbours of p
$kRNN(p)$	Set of k-reverse nearest neighbours of p, A sample q belongs to $kRNN(p)$ iff $p \epsilon kNN(q)$
d_{ij}	Distance between two points x_p and x_q
$knearestneighbourset$	$kNN(x_p)$ is defined as $\{x_q \mid d_{pq} < k^{th}\ nearest\ distance\ of\ x_p\}$, for given point x_p, the k^{th} smallest distance after sorting all the distances from x_p to the remaining points is the k^{th} nearest distance of x_p
$kreversenearestneighbourset$	$kRNN(x_q)$ is defined as $\{x_q \mid x_p \in kNN(x_q)\}$

k reverse Nearest Neighbours ($kRNN$) defines influence around a sample in terms of neighbourhood density. Note that, in case of kNNs, for a given k value, each sample in the dataset will have at least k nearest neighbours ($> k$ in case of ties) but the $kRNN$ set of a sample could have zero or more elements. The $kRNN$ set of sample p gives set of samples that consider p as their k-nearest neighbour, for a given value of k. If a sample p has higher number of $kRNNs$ than another sample q, then we can say that p has a denser neighbourhood than q. Lesser the number of $kRNN$s, the farther apart are the samples in the dataset to q, i.e. the

neighbourhood is sparse.

According to Soujanya et al [104] defined outlier as follows: An outlier sample is a sample that has less than k number of $RNNs$. That is the cardinality of RNN set which is less than k, ($|kRNNs| < k$). Lesser the number of $kRNNs$, the more distant it is from its neighbours.

In unbalanced datasets as fewer minority class samples are available, the minority samples extremely far from minority class sample subgroups are prone to be mislabeled and degrades the classifier performance while SMOTEing. So here we define the concept called extreme outlier and eliminate them as part of data preprocessing step. As the cardinality of $kRNN$ of any minority class sample p in minority class space indicates the density around p, based on this cardinality of *krnn(p)* it can be distinguished whether the sample p is in the denser part or sparser part of the minority class space. Therefore, *krnn* approach is adopted in this work for identifying sparser points (extreme outlier) in minority class space.

4.3 Extreme Outlier Elimination using kRNNs + Hybrid Sampling

Here our basic motivation is to balance the training data distribution so that prediction of minority class is better. For this, we generate the required number of artificial minority class samples using SMOTE which generates the artificial samples by interpolation. If we use SMOTE on the entire minority class samples, minority class regions may not be emphasized well if the data is very much sparsely distributed. So there is a great need of picking the samples that are in denser regions and use only those samples for generating artificial minority class samples using SMOTE. For this, we have to eliminate the samples that are far from the minority samples; we call them as extreme outliers. When we apply existing outlier detection techniques on highly overlapped and unbalanced datasets, half of the minority samples are predicted as outliers. Eliminating half of the minority samples is not feasible as they are fewer in number compared to the majority class. The k Reverse Nearest Neighbours concept is an efficient solution for this problem. The cardinality of $kRNN$ set, of a minority class sample p, depicts the density around p in terms of nearest neighbours. Based on cardinality of $kRNNs$ the minority samples are ranked and lowest ranked samples are eliminated. The

definition of extreme outlier for minority class samples is given below:

Definition: A minority class sample p is an extreme outlier, if the cardinality of $(kRNN(p))$ set is less than $k/10$ over systematically increased k values till $D * \frac{75}{100}$. Here D is the size of the Minority class samples.

We propose to combine extreme outlier concept as a data preprocessing method for minority samples and hybrid sampling approach for balancing the data distribution.

After elimination of extreme outliers, we apply hybrid sampling approach for further emphasizing the minority class samples. This is a combination of random undersampling and synthetic oversampling. It mainly works based on determining how much percentage of minority class samples (original minority class + artificial minority class samples) and majority class samples to add to the training set such that a classifier can achieve best True Positive rate (TP_{rate}) and True Negative rate (TN_{rate}). Here, TP_{rate} is the number of minority class samples correctly classified and TN_{rate} is the number of majority class samples correctly classified. As described in eq. 3.18, $G - mean$ is used for evaluating the overall classifier performance.

Fig 4.2 shows the process of generating samples for training the classifier. Initially, minority and majority class samples are separated from the dataset and we eliminated extreme outliers in the minority class samples using the method described above. Then SMOTE was applied on new set of minority class samples for the given level of SMOTE factor. For example, if we specify SMOTE factor as 5 and input minority class samples are x, then artificial minority class samples generated after SMOTE are $5x$. Generally the choice of optimal SMOTE factor is data dependent. For the dataset under consideration (Insurance Fruad Detection) the class distribution of majority and minority class samples is 94:6. So for experiments we considered SMOTE factors of 5, 7, 9, 11 and 13. Similarly we varied the Original minority class data Percentage (OMD percentage or rate), i.e., number of original minority class samples to be added to training ranging from 0 to 75. Then majority class samples are randomly undersampled from majority class data set in such a way that class distribution for training becomes 50:50. So the training dataset is an amalgamation of artificially generated majority class samples based on SMOTE factor, original minority class samples based on OMD rate and majority class samples. In this work, we have conducted experiments on four classifiers

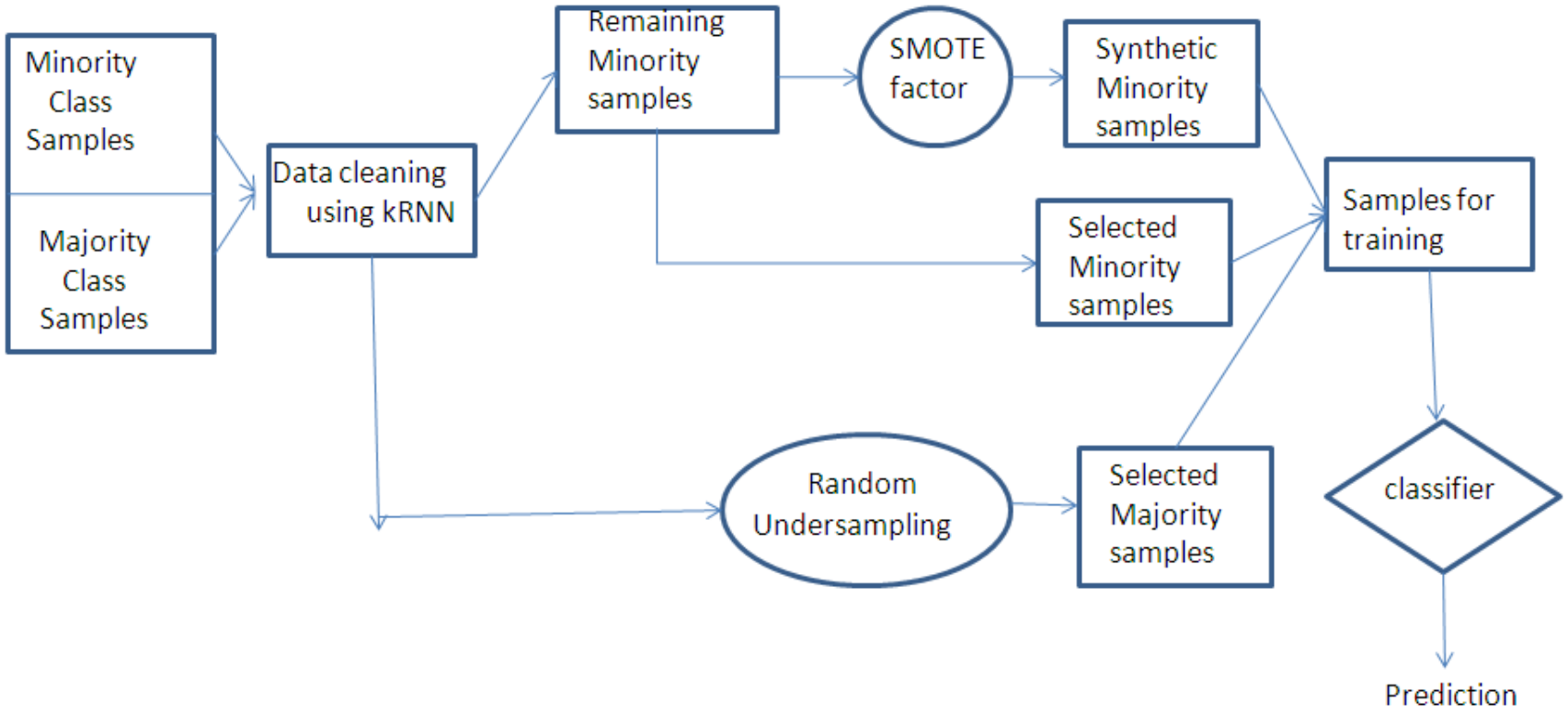


Figure 4.2: Generation of samples for training using $kRNN$ combined with hybrid of SMOTE+ RUS

namely Decision Tree (DT), Naïve Bayes (NB), k-Nearest Neighbour (kNN) and Radial Basis Function Networks (RBF).

4.4 Case study with Insurance Fraud Dataset

Experiments are conducted on an insurance fraud dataset [95]. Fraud is pervasive in today's society. Fraud poses a big problem for many industrial domains like credit card, insurance and telecommunications. Fraud detection involves identifying fraud as quickly as possible once it has been perpetrated. The challenge in fraud detection lies in the dynamic nature of the data, i.e., fraud is continually changing and constantly surviving. Thus fraud is done in various ways and modus operandi are different from one another. By using massive amounts of data (e.g. on financial transactions), we can identify the patterns of fraud using data mining methods [89]. Fraud detection poses some technical and practical problems for data mining - the most significant technical problem is due to limitations, or poor quality, of the data itself. Another crucial technical dilemma is due to the highly unbalanced nature of data of fraud detection. Typically, there are many more legitimate than fraudulent samples. Negative consequence of this type of data is higher chances of overfitting.

Even though fraud detection is a binary classification problem, in reality it is a n-class problem as each fraud is different from the other. In this chapter, we consider fraud detection as highly overlapped unbalanced data classification problem where non-fraud samples heavily outnumber the fraud samples.

4.4.1 Related Work on Fraud Detection

In this section a brief review of existing approaches for fraud detection problem are presented. An excellent survey of existing challenges in fraud detection for the different types of large datasets was done by Phua [96, 89]. This work categorizes, compares and summarizes relevant data mining based fraud detection methods. It defines the professional fraudster, formalizes the main types and subtypes of known fraud and presents the nature of data evidence collected within affected industries. Stolfo et al [106, 107] outlined a meta-classifier system for detecting the fraud, by merging the results obtained from local fraud detection tools at different corporate sites to yield a more accurate global tool. This work

was elaborated in [95, 108]. They proposed a distributed data mining model. It is a scalable, supervised black box approach that uses realistic cost model to evaluate classification models. The results demonstrated that partitioning a large dataset into smaller subsets to generate classifiers using different algorithms, experimenting with fraud/non-fraud distributions within training data and using stacking to combine multiple models significantly improved cost savings. Neural data mining approach [6] uses generalized rule based association rules to mine the symbolic data and Radial Basis Function networks to mine the analog data. It has been found that using supervised neural networks to check the results of association rules increases the predictive accuracy. Wheeler and Aitken [134] have also explored the combination of multiple classification techniques. Phua et al [95] proposed a fraud detection method, which uses stacking-bagging approach to improve cost savings. Fawcett and Provost [44] have shown that patterns of fraud can be archived by generating the fraud detection systems automatically from data using data mining techniques.

4.4.2 Dataset Description

The dataset pertains to automobile insurance and it contains 15421 samples, out of which 11338 samples are from January-1994 to December-1995 and remaining 4083 samples are from January-1996 to December -1996. There are 30 independent attributes and one dependent attribute (class label). Here, six are numerical attributes and remaining are categorical attributes. The class distribution of non-fraud and fraud is 94:6, which indicates that data is highly unbalanced. We discarded the attribute *PolicyType*, because it is an amalgamation of existing attributes *VehicleCategory* and *BasePolicy*. Further, we created three attributes namely *weeks_past*, *is_holidayweek_claim* and *age_price_wsum* to improve the predictive accuracy of classifiers as suggested in [95]. So the total number of attributes used is 33. All the numerical attributes are discrete in nature and thus are converted into categorical attributes in order to compute the distance between the samples using Value Difference Metric (VDM) [105].

Attributes with two category values like *Witnesspresent*, *AgentType* and *PoliceReportFiled* have highly skewed values where majority class samples account for more than 97% of total samples. The attribute *Make* has a total of 19 possible attribute values of which claims from 5 attribute values account for almost

90% of total samples. Average number of claims per month is 430. The detailed description of each attribute is given in Appendix-A.

4.4.3 Value Difference Metric

As the automobile insurance fraud dataset is of nominal in nature, value difference metric (VDM) [105] is used as the distance measure for SMOTEing. A simple VDM defines the distance between two values x and y of an attribute a is given as

$$vdm_a(x,y) = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q = \sum_{c=1}^{C} |P_{a,x,c} - P_{a,y,c}|^q \tag{4.1}$$

where $N_{a,x}$ is the number of instances in the training set T that have value x for attribute a;

$N_{a,x,c}$ is the number of instances in T that have value x for an attribute a and output class c;

C is the number of output classes in the problem domain;

q is a constant, usually 1 or 2 indicating L_1 or L_2 norm

$P_{a,x,c}$ is the conditional probability that the output class is c given that attribute a has the value x, i.e., $P(c|x_a)$.

As can be seen from eq. 4.1, $P_{a,x,c}$ is defined as: $P_{a,x,c} = \frac{N_{(a,x,c)}}{N_{(a,x)}}$

where Na, x is the sum of Na, x, c over all classes, i.e., $N_{a,x} = \sum_{c=1}^{C} N_{a,x,c}$

Using this distance measure two values are considered to be closer if they have more similar classifications (that is, more similar correlations with the output classes), regardless of what order the values may be given in.

4.4.4 Experimental Results and Discussion

We implemented the extreme outlier detection using $kRNN$s and SMOTE in MATLAB7.0 and used Weka3-4 toolkit for experimenting with the classifiers. Weka [137] is Java-based knowledge learning and analysis environment developed at the University of Waikato in New Zealand. Initially we eliminated the extreme outliers found from the minority samples using the method described in Section 4.2.2. For the dataset under consideration, total number of minority samples is 922. So we varied k from 10, 20, 30, 50, 100, 200, 300 to 400 and found that 92 samples qualify as extreme outliers and eliminated them from the dataset.

We computed TP_{rate} and TN_{rate} by varying OFD (Original Fraud Data) percentage and SMOTE factor. Here testing of each experiment was done against entire dataset. We set k, number of nearest neighbours to be selected while doing SMOTE, as 7, 9, 11, 13 and 15 for SMOTE factors 5, 7, 9, 11 and 13 respectively.

Total experiments conducted are 100; 25 for each classifier by doing different levels of SMOTE and varying OFD rate. Our observations from the experiments conducted using the proposed extreme outlier elimination with kRNNs combined with hybrid sampling approach on four classifiers are as follows:

- DT has shown good fraud detection rate and non-fraud detection rate in all the 25 experiments and these rates increased with increase in SMOTE factor and OFD rate (Fig 4.3 and 4.4). In all the experiments, it has given above 90% fraud detection rate.

- We observed best fraud catching rate of 99.9% with kNN classifier (Fig 4.5). This may be due to the inherent use of kNN while doing SMOTE to generate the artificial fraud samples. Non-fraud catching rate of this classifier is found to be not much effective and value is below 85% (Fig 4.8).

- For NB classifier, fraud catching rate is good which is about 85%, but it has not shown much improvement with increase in SMOTE factor and OFD rate (Fig 4.5). Non-fraud catching rate of this classifier is also found to be not much effective and the value is below 80% (Fig 4.6).

- RBF has recorded good fraud catching rate of above 85% in all the experiments (Fig 4.9). This rate is increased with increase in SMOTE factor and OFD rate in most cases. For this classifier, non-fraud catching rate is also equally good (Fig 4.10).

In this work, we also have compared the results with the same set of experiments without eliminating the extreme outliers. Our observations from comparison of two methods: Method-a: $kRNN$ based extreme outlier elimination combined with hybrid sampling and Method-b: Hybrid sampling alone are as follows Fig 4.3 to 4.10 show the results obtained in the two methods. Here the dashed lines indicate the results obtained with method-a and solid lines indicate the results obtained with method-b. Each color of line represents unique SMOTE factor i.e. the number of artificial fraud samples added to the training set.

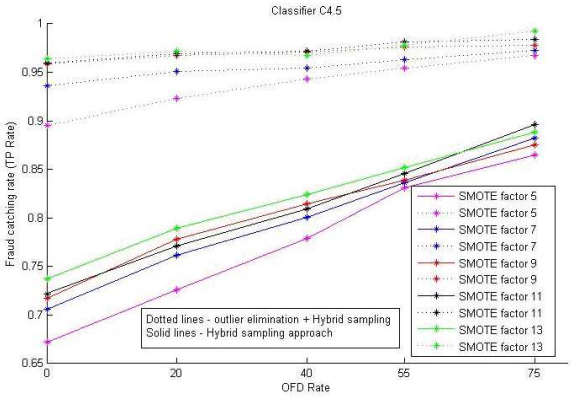


Figure 4.3: TP_{rate} Vs OFD rate for DT

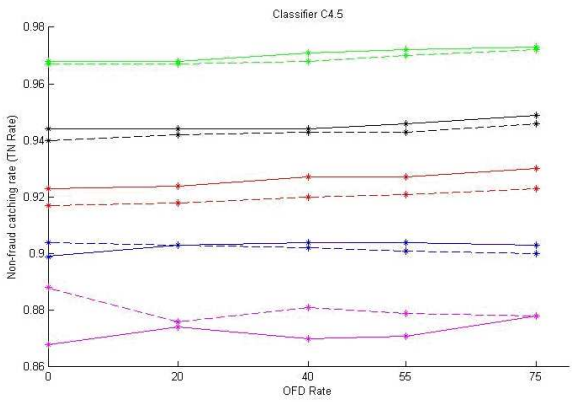


Figure 4.4: TN_{rate} Vs OFD rate for DT

Table 4.2: Comparison of Method-a with Method-b

Classifier	Method	Variation in TN rate	Variation in TP rate
DT	Method-a	87.6%-97.2%	88.8%-99.3%
	Method-b	86%-97%	67%-88%
NB	Method-a	75.8%-77.9%	84.3%-85.9%
	Method-b	75%-76%	59%-65%
kNN	Method-a	73.7%-80.8%	97.7%-99.9%
	Method-b	75%-82%	94%-99.9%
RBF	Method-a	89.4%-96.2%	86.5%-98.3%
	Method-b	90%-96%	86%-96%

*Note:***Method-a**:Extreme outlier elimination+hybrid sampling
Method-b:Hybrid sampling alone.

The $G-mean$ results which reflect overall classifier performance also reported that proposed outlier filtering with hybrid sampling over decision tree at SMOTE factor 13 outperforms the simple hybrid sampling. Table 4.3 depicts the $G-mean$ results for SMOTE factor 5 T0 13. The $G-mean$ results for RBF and kNN classifiers on proposed approach are good but not better than DT classifier.

Proposed approach is also compared with base classifier with original data (OD), random undersampling (RUS) and random oversampling (ROS) approaches.

Results on test set (see Table 4.4) indicated that the classification models learned on original data (OD) resulted in poor TP_{rate} in identifying fraud samples and poor $G-mean$ scores on overall four classification model performances. The

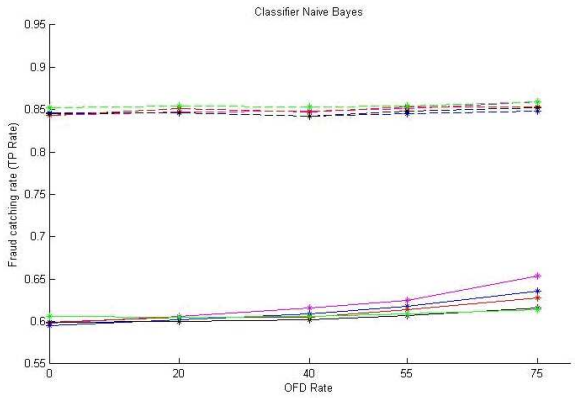


Figure 4.5: TP_{rate} Vs OFD rate for NB

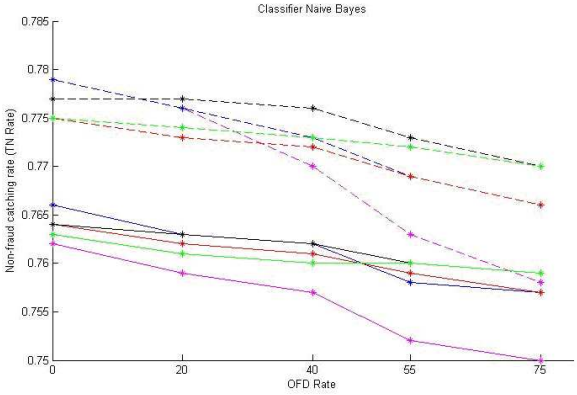


Figure 4.6: TN_{rate} Vs OFD rate for NB

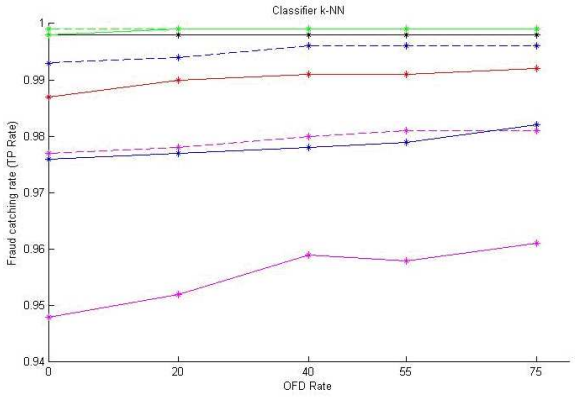


Figure 4.7: TP_{rate} Vs OFD rate for kNN

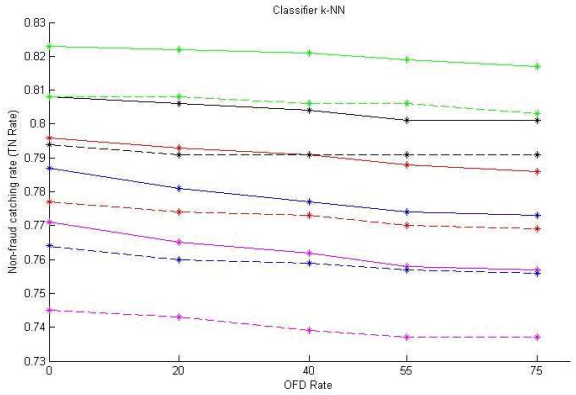


Figure 4.8: TN_{rate} Vs OFD rate for kNN

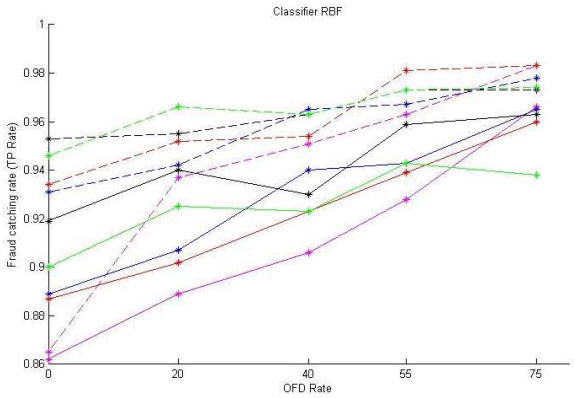


Figure 4.9: TP_{rate} Vs OFD rate for RBF

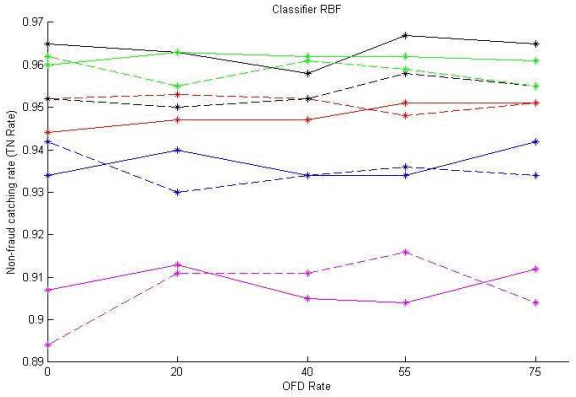


Figure 4.10: TN_{rate} Vs OFD rate for RBF

Table 4.3: Comparison of $G-mean$ across Method-a and Method-b for different SMOTE factors

SMOTE%	Classifier	Method-b					Method-a				
	OFD%	0	20	40	55	75	0	20	40	55	75
SF5	DT	0.763	0.796	0.823	0.850	0.871	0.891	0.899	0.911	0.915	0.921
	NB	0.675	0.6781	0.682	0.685	*0.700*	0.812	0.812	0.808	0.806	0.804
	RBF	0.884	0.900	0.905	0.915	0.938	0.879	0.923	0.930	0.939	0.942
	kNN	0.854	0.853	0.854	0.852	0.852	0.853	0.852	0.851	0.850	0.8502
SF7	DT	0.796	0.829	0.850	0.869	0.892	0.919	0.926	0.927	0.931	0.935
	NB	0.675	0.677	0.681	0.684	0.690	0.811	0.810	0.808	0.807	0.807
	RBF	0.911	0.923	0.936	0.938	0.953	0.936	0.935	0.949	0.951	0.955
	kNN	0.876	0.873	0.871	0.870	0.871	0.871	0.869	0.869	0.868	0.867
SF9	DT	0.813	0.847	0.869	0.881	0.902	0.937	0.942	0.9451	0.948	0.950
	NB	0.747	0.678	0.678	0.682	0.689	0.809	0.809	0.809	0.808	0.808
	RBF	0.915	0.924	0.934	0.944	0.955	0.942	0.952	0.972	0.964	0.966
	kNN	0.886	0.886	0.885	0.883	0.883	0.873	0.875	0.874	0.873	0.872
SF11	DT	0.825	0.853	0.873	0.894	0.922	0.949	0.955	0.957	0.961	0.964
	NB	0.676	0.676	0.677	0.679	0.683	0.810	0.811	0.810	0.809	0.809
	RBF	0.941	0.951	0.943	0.962	0.963	0.952	0.952	0.957	0.965	0.963
	kNN	0.897	0.896	0.895	0.894	0.894	0.890	0.888	0.888	0.888	0.888
	DT	0.844	0.873	0.894	0.910	*0.929*	0.965	0.969	0.967	0.973	**0.982**
	NB	0.679	0.678	0.678	0.680	0.682	0.811	0.811	0.811	0.811	**0.811**
SF13	RBF	0.929	0.943	0.942	*0.952*	0.949	0.953	0.960	0.961	**0.965**	0.964
	kNN	0.906	0.906	0.905	**0.904**	0.903	0.898	0.898	0.897	*0.897*	0.895

ROS approach attains the fraud catching rate (TP_{rate}) of the four classifiers DT, NB, RBF and kNN as 98%, 88.3%, 80.2% and 100%. Corresponding $G-mean$ of the ROS approach over considered classifiers improved by 78.3%, 46.7%, 57.5% and 58.3% on base classifier. The TP_{rate} of the four classifiers is 99%, 93%, 93% and 86% resepectively for RUS approach. The overall model performance $G-mean$ also increased 6.3%, 1.7%, 18% and 35.9% considerably by RUS method over base classifier. However the RUS and ROS results from Table 4.4 indicated that ROS approach overruns the RUS approach both in terms of minority class TP_{rate} and overall classifier performance $G-mean$. The drop in $G-mean$ by RUS method because of TN_{rate} compared to ROS method. Comparing with ROS method TP_{rate} of Method-a yielded superior performance approximately 1.3% and 18.1% for DT and RBF classifiers respectively and aproximately equal performance for kNN classifier. But for NB classifier the TP_{rate} degraded upto 2.4%. In terms of overall classifier performance $G-mean$ proposed approach attained superior performance upto 10% 4.6% and 22.3% for DT, NB and RBF networks respectively (see table 4.4). The improved $G-mean$ results for Method-a than ROS aproach is due to yeilding of improved TN_{rate}.

Overall, method-a performed well compared to method-b, RUS and ROS. Table 4.4 and 4.2 show the comparison of improvements in fraud catching rate and non-fraud catching rate for these methods. DT has recorded good improvement in fraud catching rate for method-a compared to method-b (Fig 4.3.) This classifier has shown good fraud catching rate of 90% at less SMOTE factors and OFD rates. Naïve Bayes classifier has suffered from low fraud catching rate in method-b (Fig 4.5). But with the proposed approach, it has recorded high fraud catching rate. Moreover for this classifier there is a reasonable improvement of non-fraud catching rate with method-a (Fig 4.6). We observed that kNN performed slightly well for method-b (Fig 4.7 and 4.8), however there is not much reduction in performance. RBF has also shown good improvement in fraud catching rate with our approach compared to method-b (Fig 4.9), but there is not much impact on non-fraud catching rate (Fig 4.10). Figures from 4.3 to 4.10 indicate that the behavior of each classifier is the same for method-a and method-b with variation of SMOTE factor and OFD rate. Further, our approach is better than RUS and ROS methods. Overall, on constructed models decision tree classifier with proposed approach yielded fraud catching rate TP_{rate} of 99.3% and effectiveness of the whole classifier

Table 4.4: Test set results across Original Data (OD), RUS and ROS

Classifier	OD			RUS			ROS		
	TP_{rate}	TN_{rate}	$G-mean$	TP_{rate}	TN_{rate}	$G-mean$	TP_{rate}	TN_{rate}	$G-mean$
DT	0.036	1	0.189	0.993	0.064	0.252	0.98	0.966	0.972
NB	0.091	0.982	0.298	0.93	0.107	0.315	0.883	0.664	0.765
RBF	0.028	0.997	0.167	0.93	0.13	0.347	0.802	0.688	0.742
kNN	0.101	0.998	0.317	0.863	0.531	0.676	1	0.81	0.9

$G-mean$ of 98.2 (see tables 4.2 and 4.3) which is superior over all other constructed models for automobile insurance fraud dataset.

In order to provide detailed comparative study across considered sampling methods, obtained $G-mean$ results are ranked across different sampling algorithms by Friedman's ranking method as described in [37] (see Table 4.5). The detailed descrption of Friedman's ranking is discussed in chapter 3. Among all methods that are considered, proposed $kRNN$ based hybrid received best mean rank of 4.5 using Friedman's ranking. Hybrid of SMOTE+RUS and ROS received the scond best mean rank of 3.75. The other methods RUS and OD received inferior ranks.

Further the statistical significance of the proposed approach relative to other sampling methods is measured with paired $T-Test$, which verifies whether the average difference between the classifier performances is significantly differant from zero. Here the paired $T-Test$ reveals that proposed approach is statistically better than RUS and OD on zero median hypothesis which is rejected at 5% significant level respectively.

4.5 Chapter Summary

This chapter introduced a new approach for eliminating outliers from the highly unbalanced datasets. In this work we defined the concept called *extreme outlier* and used k Reverse Neighbours ($kRNN$) to find them. Results show that extreme outlier elimination combined with hybrid sampling can improve the accuracy of the classifier for minority class. Here we used SMOTE to artificially create fraudulent samples and emphasize the fraud regions after elliminating extreme outliers in the fraudulent samples are eliminated. Experiments are conducted on publicly available insurance fraud dataset for four classifiers namely DT, NB, kNN and Radial Basis Function networks. The results obtained indicate that the proposed approach is efficient for fraud detection. Thus intelligent use of $kRNN$ for extreme outlier elimination of fraudulent samples and SMOTE for generating artificial minority samples resulted in improving the fraud catching rate and non-fraud catching rate. Though our approach is implemented for insurance domain, it can be applied to other domains as well for fraud detection.

Table 4.5: Ranking of classifiers based on $G - mean$ across experiments with Original Data(OD),RUS and ROS

Classifier	OD	RUS	ROS	SMOTE+RUS	kRNN+SMOTE+RUS
DT	1	2	4	3	5
NB	1	2	4	3	5
RBF	1	2	3	4	5
kNN	1	2	4	5	3
AVG.Ranking	1	2	3.75	3.75	4.5

Informative sampling - undersampling

Chapter 5

Majority Filter based Minority Prediction: (MFMP)

This chapter proposes a new informative undersampling technique for improving the minority class prediction in case of unbalanced datasets. The rest of the chapter is organized as follows. Background for Majority Filter-based Minority Prediction (MFMP) is presented in section 5.2. Section 5.3 describes proposed MFMP approach. We conducted an experimental study on three UCI repository [9] data sets and one synthetic dataset. Section 5.4 presents the automatic cluster counting approach for identifying appropriate number of clusters in minority space. Section 5.5 provides the results and discussion based on the experimental study. In section 5.6 chapter summary is provided.

5.1 Introduction

From the experimental studies of Batista et al [5] and Japkowitcz and Stephen [63] it is proved that the data set characteristics like size of the dataset, degree of overlap of minority class with majority class as well as minority class small disjuncts [65] that are small clusters without enough points, makes it harder for the classification algorithm to identify the minority class from majority class data. Therefore, conventional classification algorithms fail to predict from unbalanced data sets.

Weiss et al [130] and Estabrooks et al [40] proved that the natural distribution of majority and minority is not best for learning. Identification of best learnable

distribution from the natural distribution is needed for better prediction. Resampling techniques like undersampling and oversampling techniques are used to obtain balanced distribution for better prediction. Random undersampling and random oversampling are most popular ones. The former leads to loss of majority class information and the latter causes over-fitting to minority class. The choice of applying random undersampling or oversampling is data driven [40]. Drummond and Holte showed that random undersampling yields better minority prediction than random oversampling [35]. However, they suggested that random undersampling leads to loss of data, which is a problem. To address the problem of minority prediction in unbalanced data sets, we propose an undersampling approach called Majority Filter-based Minority Prediction (MFMP). In this approach, an unsupervised learning technique, Partition Around Medoid clustering algorithm [53] is adopted for selecting the majority class data for improving the prediction of minority class data. To avoid data loss from majority class due to undersampling, random undersampling is used along with individual clusters.

5.2 Background

Several informative undersampling methods were discussed in chapter 2.2.1. They are of noise filters [117, 136, 150], Condensation algorithms [70] and prototype selection methods [29, 146]. In this chapter a more focused undersampling based on PAM clustering is discussed for improving the classifier performance. In every iteration of the proposed MFMP algorithm, increased number of clusters and their medoids and radius are computed for distinguishing minority class regions from outnumbered majority class samples. In the following section a little background on partition around medoid (PAM) algorithm is presented.

5.2.1 Partition Around Medoid (PAM) Clustering Algorithm

PAM clustering algorithm [53] is a heuristic clustering algorithm that partitions the data around the user specified k-number of medoids (See Algorithm 3). Initially the algorithm starts with randomly chosen k-medoids. Points are assigned to their closest medoids and the total cost of making the k-number of clusters is computed. The algorithm continuously replaces the old-medoids with the newly

selected medoids until the cluster alignment leads to minimum cost. Due to the robustness towards handling noise the unsupervised learning technique PAM has been used for identifying the number of minority clusters in our proposed algorithm.

Algorithm 3 PAM Clustering Algorithm (taken from [54])

Input:
D=Dataset of N objects;
k=Number of Clusters;
Sim=Function for similarity (or Disatance) measure;
Output:
A set of k clusters;
Begin
Compute the similarity matrix for database D using given similarity (or distance) function Sim;
Randomly choose the k objects as initial set of medoids;
repeat
  Assign each non-selected object to the cluster with the nearest medoid;
  Randomly select a non medoid object, mark it as $O_{nonmedoid}$;
  Calculate the total cost, C, for swapping o_j with $O_{nonmedoid}$, where o_j belongs to the cluster representative of the current medoid;
  if $C < 0$ **then**
    swap o_j with $O_{nonmedoid}$ to form a new set of medoid;
  end if
until there is no change in cost C

5.3 Majority Filter-Based Minority Prediction (MFMP)

The goal of our approach is to achieve good prediction over minority class and avoiding unnecessary information loss from the majority class [40, 35]. To achieve this goal, we use both selective sampling and random sampling of the majority class samples as shown in the workflow diagram in Fig 5.1. The justification for adopting of both selective sampling as well random sampling is as follows:

Selective sampling is for attaining good prediction over the minority class. Since any overlap between majority and minority classes hampers minority prediction, selecting majority samples that are far from the minority decision regions may be beneficial. An unsupervised learning technique, Partition Around Medoid Clustering algorithm (PAM) (section 5.2.1) is adopted for identifying the maximum number of minority clusters in the data. The remaining majority class samples that are not part of the minority class decision regions then are selected. The newly selected majority class samples bring good separation between majority and minority class samples, thus the classification algorithm easily learns from the minority class.

The random selection of samples on individual clusters is for attaining good prediction over the majority class. In this step we randomly pick some of the points from individual clusters without disturbing the minority class prediction attained by the MFMP cluster approach. Proposed Majority Filter-based Minority Prediction (MFMP) algorithm is described as follows.

Let s_{min} be a bin of minority class samples. Since no prior knowledge is available about how many minority clusters exist in the data set, the proposed approach initially starts with two clusters over s_{min}. Once the initial minority clusters are formed, cluster centroids are used as the representatives for the whole minority dataset. Majority class samples that fall within the maximum radius of the minority cluster are assigned to its nearest minority cluster. Remaining majority samples that are not inside of the minority clusters are selected and stored in $Current\_Selected\_maj$ bin. The cluster number is incremented for the next iteration. In Algorithm 4, $Current\_Selected\_maj$ variable holds the number of majority samples being selected in every iteration. $Old\_Selected\_maj$ variable holds the selected samples of the previous iteration. The variable $Filter\_Change$ holds the difference between last $Old\_Selected\_maj$ and $Current\_Selected\_maj$ variables. The algorithm terminates when $Filter\_Change$ is negligible considerably over (set to a threshold Δ) the last $Old\_Selected\_maj$ and current $Old\_Selected\_maj$ variable updates. The threshold Δ ensures avoiding in forming of small or singleton clusters and reflects the current cluster alignment. At particular point in MFMP procedure, further increment in cluster count does not show much improvement in $Filter\_Change$ variable. Once the maximum number of clusters is formed, clusters having singleton minority class element are labeled as outliers and eliminated.

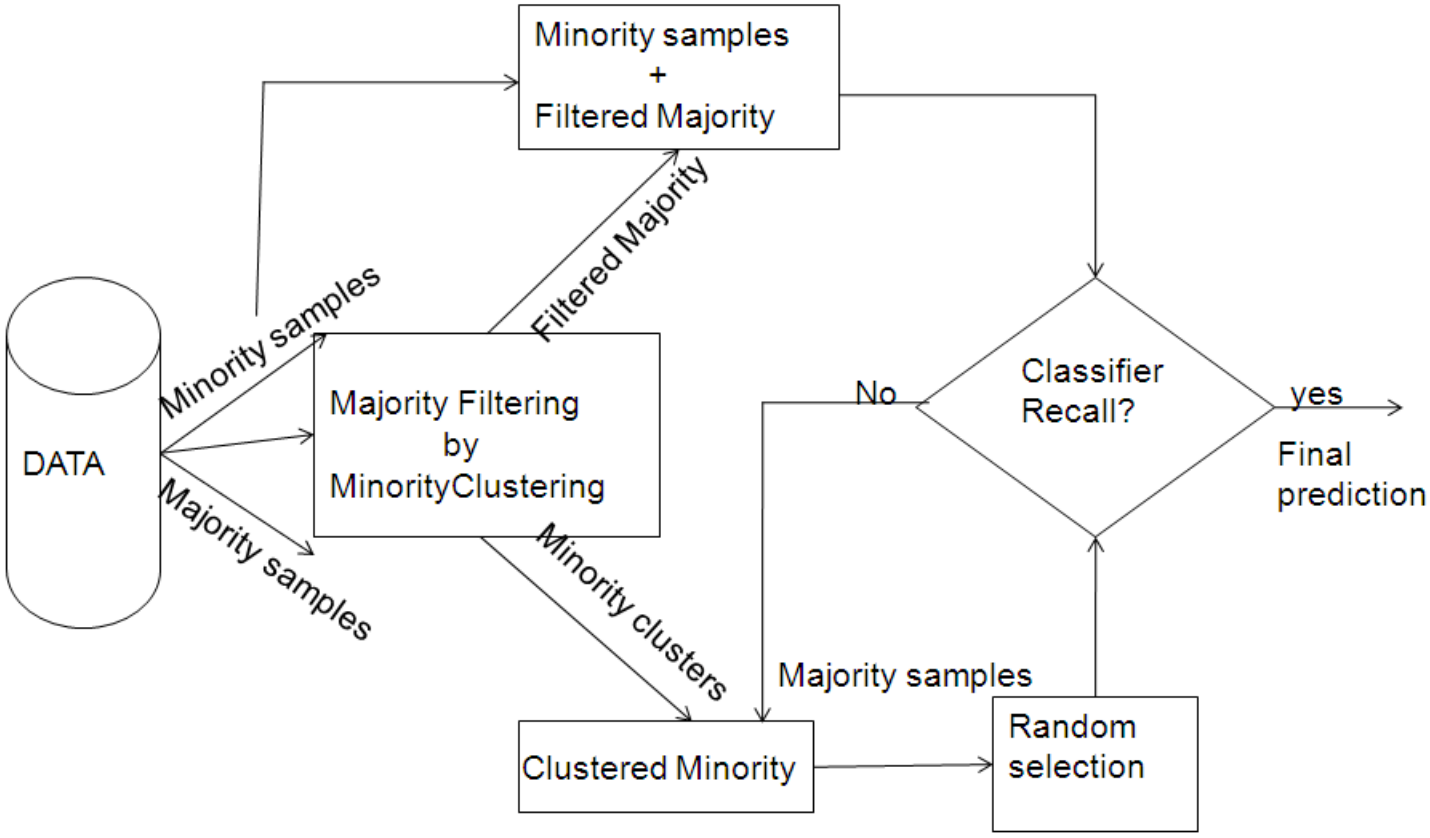


Figure 5.1: Flow diagram for MFMP.

Algorithm 4 Proposed MFMP Algorithm

Input:

s_{min}=Minority class samples; s_{maj}=Majority class samples;

Δ=user input;

Output:

S_total=New trainingset; k_{min}=Number of minority class clusters;

Variables:

$k_{min}\_medoid[]$=k number of minority class medoids; $k_{min}\_radius[]$=radius of each minority class cluster;

C_s_{min}=minority class points in cluster C; C_s_{maj}=Majority class points in cluster C;

Begin

k_min=2, Current_Selected_maj=0; Old_Selected_maj=0; Filter_change=size(selected_maj);

repeat

Compute k_{min} number of clusters of s_{min} and obtain k_{min}_medoid medoids;

Compute distance $dist\_s_{maj}$, from all k_{min} medoids to the points in s_{maj};

Old_Selected_maj= Current_Selected_maj;

Selected_maj=$dist\_s_{maj} > k_{min}\_radius$;

$Filter\_chang = sizeof|Current\_Selected\_maj - Old\_Selected\_maj|$;

k=k+1;

until $Filter\_chang > \Delta$

$S_{total} = s_{min} + Current\_Selected\_maj$;

for each cluster **do**

Compute Imbalance Ratio (IR)= $\frac{C\_s_{maj}}{C\_s_{min}}$ of each minority class cluster;

if $IR > 0.5$ **then**

Random_select = minority class number of majority points from each cluster;

$S_{total} = S_{total} + Random\_select$;

end if

end for

End

After that $Current\_Selected\_maj$ bin is added to s_{min}. Now a classifier is learned on s_{min}. Selective sampling of those majority class samples far from minority class sub regions is for improving the minority class prediction in the given dataset.

To improve the prediction for the majority class, clusters that have imbalance ratio (ratio of majority and minority samples) more than 0.5 are selected. Now, majority samples are selected randomly from among these clusters until the selected majority points do not distort the minority *recall* rate learned by the classifier during the first step. This process improves the $precision$ and $F-measures$ for the given dataset.

5.4 RNN Curve based Cluster Counting

In this work we have investigated an alternative for finding best number of clusters in a dataset using Reverse Nearest Neighbour (RNN) concept described in 4.2.2. This approach identifies number of clusters based on RNN Curve algorithm, which is derived from the Reverse nearest neighbour concept. The RNN curve is a plot, which depicts the neighbourhood influence variations of a query point with respect to whole dataset. Korn and Muthukrishnan [68] suggest that the concept of Reverse Nearest Neighbours can be used to capture the neighbourhood influence (density) around a query point. Furthermore, according to [104] neighbourhood influence property in varying number of nearest neighbours is useful for differentiating dense clusters with outlier points in a dataset. In the proposed contribution, we followed the suggestions of Korn and Muthukrishnan [68] and Soujanya et al. [104] for counting the number of clusters for partition clustering algorithms. The details of RNN are discussed in section 4.2.2. RNN based cluster counting consists of two-steps:

- Generating Reverse nearest neighbour curve.
- Counting the number of steps in reverse neighbour curve.

5.4.1 Generating Reverse-NN Curve

The RNN curve algorithm is based on the observation that as the number of nearest neighbours increases around a query point, the number of reverse neighbours also increases. The input for RNN curve algorithm is nn_matrix of size

$n \times n-1$; which contains nearest neighbour indexes sorted according to the smallest distances. Here n is the size of the dataset. To plot RNN curve for each query point in a dataset, the algorithm searches for the query point in each column of nn_matrix. If the column contains the query point then the algorithm increments the RNN_Count with the number of times query point has appeared in the column. In case the algorithm does not find the query point in a column then RNN_Count is constant until the query point is in other column. If the query point is not present in the columns, it implies there are no neighbours to influence the query point. Here traversing each column of nn_matrix means incrementing kNN count, for $k = 2$ to N. Periodically counting RNN_Count means estimating the influence around the query point for that k value. Algorithm 5 depicts the generation of RNN curve. This algorithm captures the growth and constant variation rates of RNN_Count in RNN_Plot variable. This growth rate of the reverse neighbours count (RNN_Plot) is plotted against different nearest neighbour values (columns of nn_matrix). As shown in Fig 5.2, the resultant plot is a curve with some linear portions indicating constant change in RNN_Count and with some raising portions indicating increase in RNN_Count. This sudden growth and saturation rate of the RNN_Counts lead to stepping behavior in RNN curve. If the data set has clusters, the RNN_Count for a query point increases until it covers one cluster. After that the RNN_Count is constant until the points in other clusters start to influence the query point. So the stepping nature of the RNN curve indicates the number of clusters present in the dataset. If the curve has two linear steps means the data set has two clusters. Therefore, the number of linear portions in a RNN curve is the same as the number of clusters in the dataset.

5.4.2 Counting Clusters in Reverse-NN Curve

Once RNN curve is generated, according to Guha et al. [49] any curve exhibiting linear, exponentially increasing, and saturation behavior can be considered as sigmoidal (spline) in nature. From Fig 5.2 it is visually clear that the RNN curve is also sigmoidal in nature. The algorithm for counting clusters in RNN curve is based on identifying the slope of the sigmoidal curve [49]. Since the RNN curve is also sigmoidal in nature, the linear portions of the curve represent minima and the raised portions of the curve represent maxima. The Algorithm 6 describes

Algorithm 5 Algorithm for RNN Curve generation

```
k_min = RNN_curve_Generation(Minorityclasssamples)
Input: nn_matrix;
Output:RNN_Plot[]; /* A matrix of RNN_Count for all datapoints *\
for each Point in the database do
  RNN_Count=0; */Reverse nearest neighbour *\
  count
  point_count=0
  for each column in the nn_matrix do
    if (Point) in the nn_matrix column then
        Point_count=No.of times point appears in the column.
        RNN_Count=RNN_Count+Point_count;
        RNN_Plot[]=RNN_Count;
    else
        RNN_Plot[]=RNN_Count;
    end if
  end for
   Plot RNN_Plot[] Vs columns of the nn_matrix;
  k_min=RNN_Curve_Counting(RNN_Plot[]);
  return k_min;
end for
```

counting of the number of clusters in RNN curve. But all the points in the data set do not exhibit stepping behavior. From Guha et al. [49] the length of the

Algorithm 6 Algorithm for finding cluster count in RNN curve

k_min=$RNN\_Curve\_Counting(RNN\_Plot[])$
Input: RNN curves
Output: Maximum number of peaks
$N_{root,max}$=0;
for each Point in the database **do**
  C=RNN_Plot[Point];
  C_s=Smooth(C); /* Obtain first order derivative *\
  $C_s'' = Smooth(d^2C/dNN^2)$ /* Obtain second order derivative *\
  N_{root}=Number of maxima in C_s'' by peak picking routine;
  if $N_{root} > N_{root,max}$ **then**
    $N_{root,max} = N_{root}$;
  end if
end for
if $N_{root,max}$ is even **then**
  $k\_min = N_{root,max}/2$;
else
  $k\_min = (N_{root,max} + 1)/2$;
end if
return k_min;

lower linear portions of the sigmoidal curve represents the position of the query point in the data space (outlying). Since the curve generated by RNN_Count is also sigmoidal, the length of the longer lower tail (the initial linear portion) of the RNN curve indicates that the point is located in a sparse region of the data space. A very small tail of the RNN curve represents that the point is located in dense region of the data space. Fig 5.2 and Fig 5.3 show the RNN curves with query points located in a sparse or dense region in a data space. The length of lower tail of RNN curve is calculated by taking the difference of successive RNN_Counts. The Algorithm 7 describes finding the length (outlying point) of the RNN curve. Here $RNN\_Count_{(k+1)NN}$ represents RNN_Count at $(k+1)NN$. RNN_Count_{kNN} represents RNN_Count at kNN. RNN_Length represents maximum count of the RNN curve at which the RNN_Count first starts to increase exponentially.

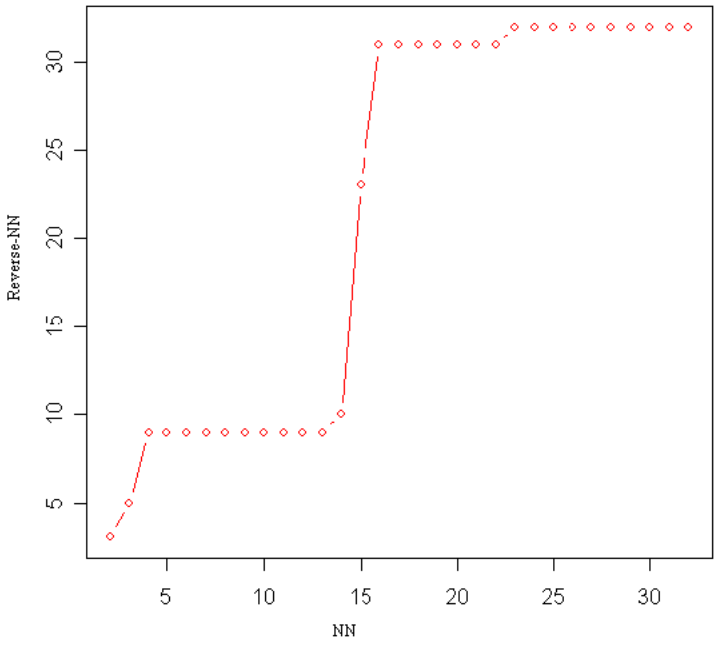


Figure 5.2: A query point located in a dense region of the data space

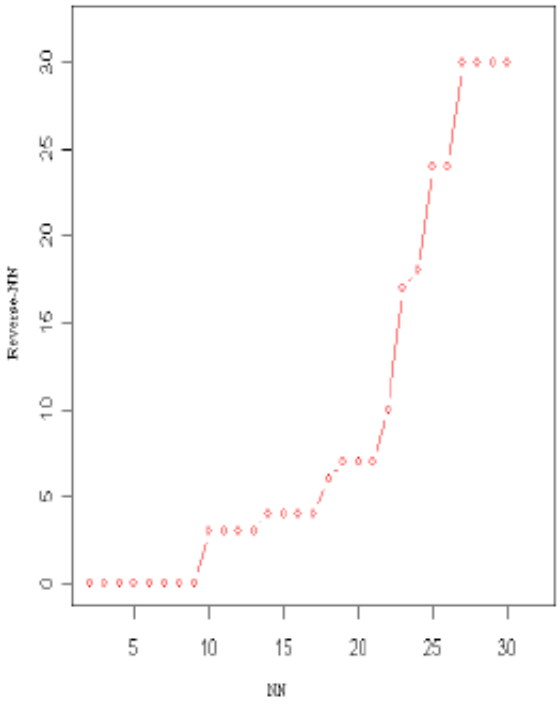


Figure 5.3: A query point in a sparse region of the data space

Algorithm 7 Algorithm for finding the length of RNN curve

$RNN\_Length = RNN\_Curve\_Length(RNN\_Plot[])$
RNN_Length=0;
for k= 1 to N **do**
  $L_{max} = RNN\_Count_{(k+1)NN} - RNN\_Count_{kNN}$;
  RNN_Length=RNN_Length+1;
  if $L\_max > 0$ **then**
    break from **for**
  end if
end for

For any RNN curve, if RNN_Length in Algorithm 7 is at small kNN value for $k = 1$ to N, then the point is located in denser portion of the data space, else it is located in a sparse portion of the dataset. Algorithm 8 depicts MFMP algorithm with RNN based cluster counting approach for a priori prediction of the best number of clusters in minority class

5.5 Experimental results

We evaluated the proposed approach on one synthetically generated and three UCI repository datasets. The datasets that are used for the evaluation of MFMP are described in Table 5.1. Here, for each dataset, number of examples, number of attributes, class Label description (whether it is majority or minority class), class percentage of class distributions and number of minority clusters obtained through RNN are depicted. The following data sets are considered for experimentation: $Satimage$, E_Coli datasets and $Haberman$ datasets all taken from the UCI repository. In all these datasets the minority class is skewed (degree of overlap) with respect to majority class. As $Satimage$ and E_Coli data sets have more than two classes, the data sets are converted into binary class datasets. This is done by considering the class with fewer samples as minority class and rest of the samples as majority class, as suggested by Batista et al [5].

The $Haberman$ dataset is two-class data set about breast cancer patients whether they survive or die within 5 years period of surgery. Furthermore, a synthetic data set is generated based on random number generation technique. All attributes in the considered datasets are quantitative. We implemented MFMP in

Algorithm 8 MFMP Algorithm based on automatic cluster counting using RNN

Input:

s_{min}=Minority class samples, s_{maj}=Majority class samples;

Output:

S_total=New trainingset, k_{min}=Number of minority class clusters;

Variables:

k_{min}=number of minority class medoids; $k_{min}\_radius[]$=radius of each minority class cluster;

C_s_{min}=minority class points in cluster C; C_s_{maj}=Majority class points in cluster C;

Begin

$k\_min = RNN\_Curve\_Generation(Minority\ class\ samples)$,

Compute k_{min} number of clusters on s_{min} and obtain k_{min}_medoid medoids;

Compute distance $dist\_s_{maj}$, from all k_{min} medoids to the points in s_{maj};

Selected_maj=$dist\_s_{maj} > k_{min}\_radius$;

$S_{total} = s_{min} + Selected\_maj$;

for each cluster **do**

  Compute IR= $\frac{C\_s_{maj}}{C\_s_{min}}$ of each minority class cluster;

  if $IR > 0.5$ **then**

    Random_select[] = minority class number of majority points from each cluster;

    $S_{total} = S_{total} + Random\_select[]$;

  end if

end for

End

Table 5.1: Dataset Description

Dataset	No.of Examples	No. of. Attributes	Min-Maj Label	Maj%:Min%	No.of. Minority Clusters from RNN curve
Simulated	329	2	Neg-Pos	89:11	4
Satimage	6435	36	Remainder-4	90:10	4
E_Coli	336	7	Remainder-iMU	89:11	6
Haberman	306	3	Survive-Die	74:26	5

MATLAB 7.0. For studying the behavior of MFMP, the following classifiers were used: DT, kNN, NB, and RBF. The classifiers were as implemented in Weka-3 4 [137] toolkit. For experimental purpose threshold Δ is set to final minority cluster alignment k value.

As described in section 3.2.1, we used minority class *precision*, *recall* and $F-measure$ to evaluate the experimental results of MFMP. This was performed by comparing Original Unbalanced Data OD and RUS with MFMP. In this context, OD approach is a classifier learned on original dataset and RUS is a classifier learned by random undersampling of the majority class. The best number of clusters obtained through RNN curve approach is depicted in 5.1.

Fig 5.4 to Fig 5.7 depict the behavior of the considered approaches on four different classifiers DT, kNN, RBF and NB respectively. Each figure corresponds to relative comparison of the MFMP with OD and RUS over four considered datasets on a single classifier. We have used dataset numbers 1, 2, 3 and 4 for referring the data set names in the graphs. The results obtained are by taking average of 10-fold cross validation on each classifier.

It is clear from Fig 5.4 to Fig 5.7 that the classifiers learned by the MFMP attain high *recall*, good *precision* and considerable $F-measure$ rates than the classifiers learned using the OD and RUS approaches. For DT classifier, the MFMP approach attains 95% *precision*, *recall* and $F-measure$ rates on *Satimage* dataset. On E_Coli dataset it has achieved 46% *precision*, 97% *recall*, 63% $F-measure$ rates. On $Haberman$ dataset it attains 31% *precision*, 99% *recall*, 49% of $F-measure$ rates and 57% *precision*, 88% *recall*, 69% $F-measure$ on synthetic data set. On RBF network MFMP has reported, best $F-measure$ of 61% on E_Coli data, 54% of $F-measure$ on synthetic dataset, 45% on $Haberman$ dataset and 34% on *Satimage* dataset. On NB classifier the MFMP attained best $F-measure$ at 54% for Satimage dataset. E_Coli dataset attained 53%,44% for $Haberman$ dataset and 34% of $F-measure$ for synthetic dataset. For kNN classifier the proposed approach achieves best $F-measure$ of 95% on *Satimage* dataset. On E_Coli dataset it attains 55%, 46% On $Haberman$ dataset and 44% on synthetic dataset.

By comparing the MFMP results with other approaches studied here, we observed the following. From Fig 5.4 to Fig 5.7, *recall* rate for the classifiers learned by the random undersampling approach is considerably higher and the *precision*

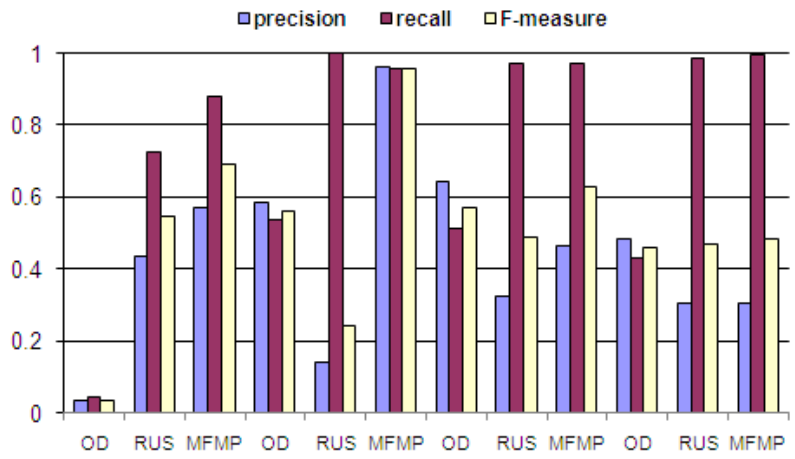


Figure 5.4: Results of Decision Tree Classifier

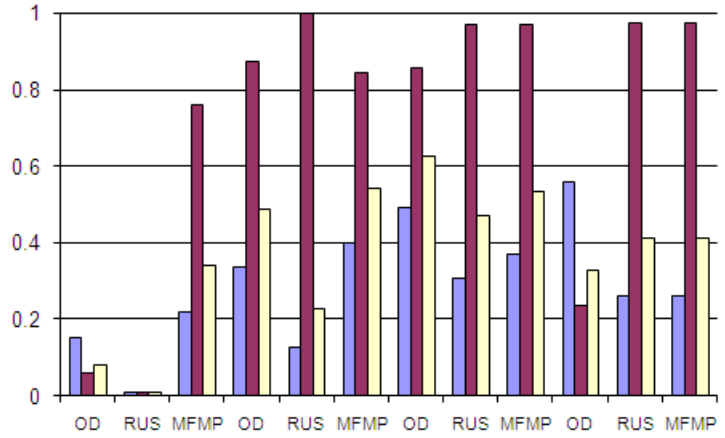


Figure 5.5: Results of Naïve Bayes Classifier

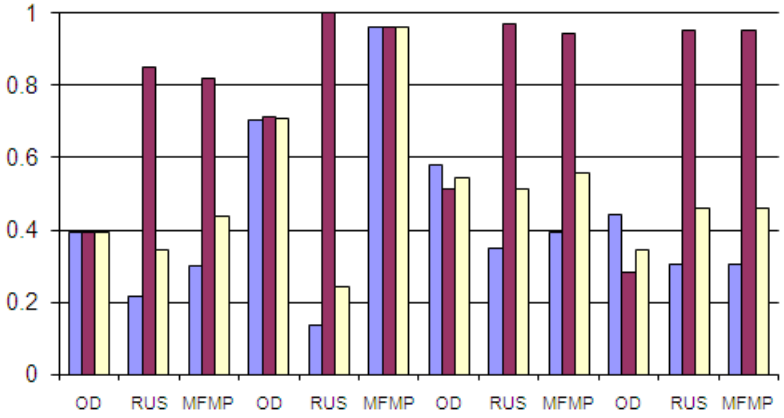


Figure 5.6: Results of k-Nearest Neighbour Classifier

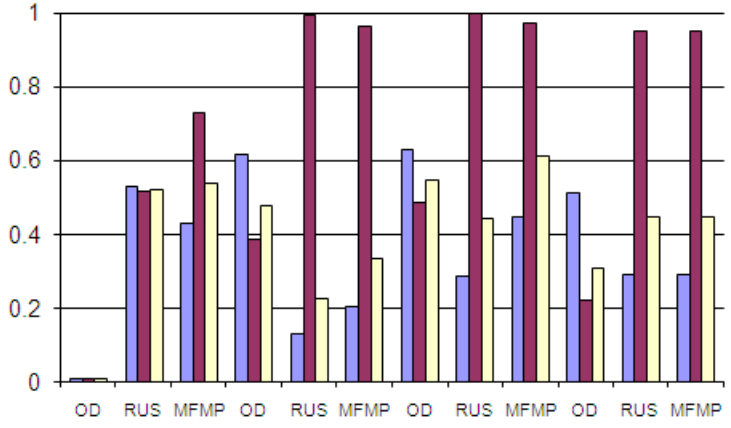


Figure 5.7: Results of Radial Basis Function Network Classifier

rate is significantly lower. This indicates that the classifier perhaps lost the valuable information from the majority class in order to bring good prediction over the minority class. Whereas the classifiers learned from the original data (OD) exhibit high *precision* rates and low *recall* rates showing less prediction rate for the minority class. We have ranked the classifiers based on their prediction rate ($F - measure$) obtained in the MFMP approach. Among all classifiers and on all data sets the decision tree classifier outperforms the remaining classifiers. For the *Satimage* dataset the prediction rate is 95% and the dataset exhibits cluster behavior for the minority class. The kNN classifier stands at second place. The RBF and NB are at the third place. This ordering may be due to unique generalizing capabilities of various classifier. From the experiments we identified that for moderately unbalanced class distributions where minority class represents low concept complexity (overlapping), our proposed approach exhibits good performance.

5.6 Chapter Summary

Since most of the datasets in real world applications are unbalanced, resampling techniques are one of the popular techniques to solve this problem. However, they suffer from information loss from the majority class. In order to solve this problem, we propose a Majority Filter based Minority Prediction (MFMP) approach for selecting pure majority class samples. This selection procedure involves selective sampling and random sampling of the majority class for improving both *precision* and *recall*. For selective sampling of majority class that are not falling in the impure minority clusters, PAM algorithm is adopted for minority class samples. The best number of clusters is estimated through RNN curve based cluster counting approach. We tested our approach on decision tree, k-Nearest Neighbour, Naïve Bayes and Radial Basis Function networks using one synthetic and three real world datasets. For moderately unbalanced datasets where minority class exhibits cluster nature the proposed approach exhibits good *precision*, high *recall*, good $F - measure$ rates for the unbalanced datasets. Proposed approach outperformed random undersampling approach in this scenario. Decision trees gave good prediction compared to other classifiers studied here.

Chapter 6

A Probabilistic Cost Weighted Active Learning Approach

The previous chapter 5 discusses a passive method for undersampling. This chapter provides a new active learning based undersampling solution to improve the performance of Support Vector Machine (SVM) classifier performance in case of unbalanced datasets. The rest of the chapter is organized as follows. Section 6.2 discusses work related to undersampling issue in SVM classifier. Section 6.3 provides the background and motivation. Section 6.4 presents the exploration of probabilistic active learning, Statistical Query SVM (StatQSVM) on class imbalance problem. Proposed Cost Weighted Statistical Query SVM (CStatQSVM) algorithm with new stopping criterion is presented in Section 6.5. Section 6.6 discusses the experimental results. In particular, a comparative study is presented with active learning methods such as Learning on the border (LOB) and StatQSVM as well as with other conventional methods that address the class imbalance problem such as random undersampling and different cost estimation methods. Finally, the chapter is concluded in Section 6.7.

6.1 Introduction

From the experimental studies of Japkowicz et al. [62] it is proved that Support Vector Machine (SVM) classifier is less sensitive to class imbalance problem compared with rest of the global classifiers such as decision tree and Neural Networks. According to Wu and Edward [139] the imbalance training set ratio and the im-

balance support vector ratio causes skewed boundary towards the minority class, thus hampering minority class prediction. Resampling techniques are quite popular in data mining and machine learning to counter the class imbalance problem. Re-sampling techniques like undersampling and oversampling techniques are used to obtain balanced class distribution for better prediction. As the training time of SVM models scales quadratically with number of training instances, in the case of large scale training data oversampling, causes computational overhead to the predictive model by incorporating extra data, whereas the undersampling yields better minority class prediction from the SVM model. However, from Akbani et al. [1] random undersampling leads to performance degradation due to loss of informative instances from majority class. Selecting of informative instances that leads to improvement in SVM model performance is an important issue.

In this chapter we show that undersampling based on cost weighted probabilistic active learning is effective to solve this issue. The main objective behind active learning is to iteratively query for informative instances which can give rise to the model performance that resembles the performance of the entire training set. Unlike traditional query based active learning algorithms [12, 101, 116] that query a point based on the proximity to the current hyperplane, StatQSVM [86] selects set of points based on a distribution determined by the current hyperplane that models the class separation and a locally defined confidence factor.
To summarize, the main contributions of this chapter are:

- Exploring the viability of probabilistic active learning (StatQSVM) on a wide variety of unbalanced datasets is the first contribution. The exploration was done by comparing the behavior of StatQSVM with query based active learning method (LOB) [39] for establishing superiority in terms of performance. The method Learning on the Border (LOB) was proposed to solve class imbalance problem. Experimental evidence shows that the performance of StatQSVM is significantly good on moderate class imbalance settings. But at high class imbalance settings a performance-drop-down is observed in minority class prediction.

- In order to alleviate this drop-down at high imbalance settings, a probabilistic cost weighted active learning SVM (CStatQSVM) algorithm, along with a new stopping criterion as an undersampling approach is proposed.

- CStatQSVM is evaluated on several unbalanced datasets. In particular the performance was compared with LOB, StatQSVM active learning methods and other conventional methods that address class imbalance problem. Experimental evidence showed that the proposed methodology is effective as compared to the rest of the methods considered for comparative study.

6.2 Related Work

As oversampling is computationally costly on SVM classifier, several researchers proposed [17, 72, 112, 146] undersampling solutions to improve the SVM classifier performance based on prototype selection, pruning of support vectors, constructing ensembles of SVM's. These methods are either ineffective or change the orientation of original hyperplane learned on original unbalanced datasets.

It can be observed that from the systematic studies of Ertekin et al. [39] and Vlachos [127], active learning yields balanced class distribution in the early rounds without loss of informative instances from both classes. Moreover, Ertekin et al. proposed an undersampling approach called Learning on Border (LOB) based on a support vector active learning algorithm QuerySVM [116] with a margin exhaustion based stopping criterion. The margin saturation criterion suggests stopping active learning when the support vectors start to stabilize. Their proposed algorithm scales well for large datasets. In this chapter, a faster method named, probabilistic active learning StatQSVM [86] is studied for the class imbalance problem. Extending the StatQSVM we propose a probabilistic cost weighted active learning algorithm (CStatQSVM) with a new confidence based stopping criterion as an undersampling method. Proposed method is quite fast and yields better minority class prediction at the early rounds of active learning compared with other active learning methods.

6.3 Background and Motivation

This section briefly describes cost weighted support vector machines and support vector active learning that motivates the current work.

6.3.1 Different Error Cost (DEC) SVM

In eq. 6.1 the tuning parameter C determines the trade-off that a model can tolerate between maximizing the margin and minimizing the empirical error.

$$\min_{w,b,\xi_i} \frac{1}{2} w.w^T + C \sum_{i=1}^{N} \xi_i \tag{6.1}$$

$$subject\ to \begin{cases} \forall i\ y_i(w^T \Phi(x_i) + b) \geq 1 - \xi_i \\ \forall i\ \xi_i \geq 0 \end{cases} \tag{6.2}$$

In case of unbalanced data if C is not large enough, in order to maximize the margin SVM tends to make zero error from majority class. The trade-off in terms of error is solely for few minority samples. Thus to improve the misclassification of minority class high predictive cost is assigned to samples from this class. To balance the misclassification cost Veropoulos et al. [126], and Morik et al. [87] introduced two loss functions C^+ and C^- for two types errors from minority and majority class, reflecting their importance during training. Therefore, the optimization problem formulation has two loss functions for two types of errors.

$$\min_{w,b,\xi} \frac{1}{2} w.w^T + C^+ \sum_{i|y_i=1}^{N^+} \xi_i + C^- \sum_{j|y_j=1}^{N^-} \xi_j \tag{6.3}$$

$$\forall k : y_k[w.x_k + b] \geq 1 - \xi_k,\ where\ \xi_k = \max(0, 1 - y_k[w_k.x_k + b]) \tag{6.4}$$

Here N^+ and $N^-$are number of minority and majority class samples in the training set. Furthermore, Morik et al. [87] suggested to set C^+ and C^- in order to satisfy the imbalance ratio, $\frac{Number\ of\ Majority\ class\ samples}{Number\ of\ Minority\ class\ samples}$ of the whole dataset, which converts into a higher weightage for minority class misclassification rate. However, tuning the loss functions to imbalance ratio balances the trade-off between minority class misclassification rate and majority class misclassification rate by pushing the learned boundary more towards the majority class and learning more room for generalization of minority class. Thus, the minority class prediction can be improved.

6.3.2 Active Learning

Usually training an SVM classifier requires significant amount of training time and huge memory space for dealing with large datasets and for solving complex

Quadratic Programming problems. Several researchers proposed active learning based solutions to reduce the computation time. Apart from improving computational ability of the underlying model, active learning is also widely used in semi supervised learning to limit the labeling effort. Recently, Bloodgood et al.[10], adopted QuerySVM [116] active learning for natural language processing tasks. The main motivation for active learning algorithms is to select informative instances while maintaining an equal performance as with the original training set. These informative instances are selected from training set through a specific iterative querying process initiated by SVM classifier.

Tong and Koller [116] presented three active learning methods for SVM classifier based on the version space reduction criterion. The authors assume that informative instances that are close to the current hyperplane divide the version space into two equal parts. Among the three proposed methods the first one QuerySVM is famous in active learning literature due to its simplicity and computational ability. QuerySVM iteratively queries for a single point that is nearer to the current hyperplane from the rest of the training data. Campbell and Cristianini [12] and Schohn and Cohn [101] proposed active learning strategies similar to QuerySVM. Moreover, Schohn et al. [101] proposed a greedy optimal strategy by calculating class probability and expected error where they have selected the training samples based on the proximity to the current hyperplane.

Rather than selecting single point based on proximity to the hyperplane, StatQSVM [86] selects q number of points by estimating a likelihood that they belong to the actual support vector set. Fig. 6.1 depicts the criterion of sample selection for StatQSVM.

From Fig 6.1. $f(x) = \langle w, b \rangle$ is the current hyperplane with the support vectors S that are represented in triangled points. The points with circles represent selected q points. These q points are again used along with the current support vectors $S = S \cup q$ to obtain new support vectors set. The likelihood of each selected point in q that it belongs to actual support vector set S is estimated by the degree of confidence that current set of SVs produce actual SVs with respect to the margin $f(x) = \langle w, b \rangle$. The confidence factor is estimated based on the local properties of the current support vectors S through a test set T.

Confidence factor estimation: According to Mitra et al.[86] degree of confidence defines the degree of closeness to the actual support vector set to current

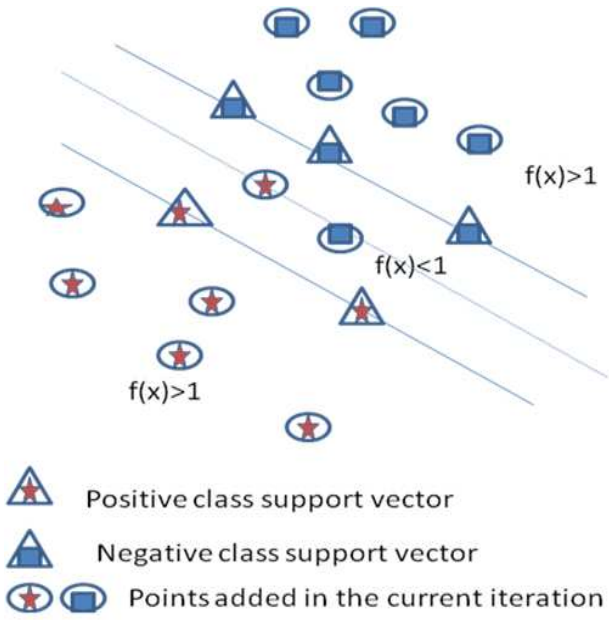


Figure 6.1: Sample selection criterion for StatQSVM. Where $f(x) = \langle w, b \rangle$ is the current hyperplane with support vectors S (the triangled points), and with the newly queried q points that are added in every iteration (the circled points).

support vector set. Let $S = \{s_1, s_2, s_3, ...s_l\}$ be current support vector set with l support vectors, $T = \{t_1, t_2, t_3, ...t_n\}$ be the test set. For each of the support vectors, k nearest neighbours are computed where $k = \sqrt{l}$. Among the computed neighbours if k^+ and k^- are the minority and majority class nearest neighbours, then the confidence factor c is estimated as [86],

$$c = \frac{2}{lk} \sum_{i=1}^{l} min(k_i^+, k_i^-) \tag{6.5}$$

Here k_i^+, k_i^- are the minority and majority class nearest neighbours for support vector i. Theoretically, the value of c ranges between 0 and 1. Maximum value of $c(= 1)$ occurs, when $k_i^+ = k_i^-$, indicating that the current support vector set might represent the actual support vectors. The minimum value of $c(= 0)$ indicates that the current support vector set is far from the actual boundary. The higher the value of c the closer is the current support vector set to the actual support vector set. Yielding maximum value for c purely depends on how the test set samples are distributed with respect to the current support vectors set S.

6.4 Exploratory Study of StatQSVM

In the literature, there is no systematic study of the application of an active learning approach such as StatQSVM for unbalanced datasets. So initially we performed an exploratory study to explore the viability of StatQSVM algorithm on class imbalance problem. This section describes the experimental setup with datasets being used for empirical evaluation of StatQSVM and discusses the results obtained.

6.4.1 Datasets

Nine datasets from the UCI repository [9] were used for the experimentation. The characteristics of the datasets are shown in Table 6.1. Out of these datasets, Ionosphere, Pima and Magic Gamma Telescope (Gamma) datasets were meant for two-class classification problem.

- *Ionosphere* data set describes the classification of radar returns from ionosphere into good or bad returns. The 'good' radar returns show the evidence for a type of structure in ionosphere whereas 'bad' return does not show the evidence.

- The *Pima* data set is about predicting whether a patient has diabetes (1) or not (0).

- The *Magic Gamma Telescope (Gamma)* data set describes the discrimination of primary gamma signals (g) from those images generated by cosmic rays (h) in the upper atmosphere.

The rest of the datasets (*Glass*, *Waveform*, *Letter-a*, *Satimage*, *Abalone* and *Shuttle*) have more than two classes. The datasets are converted into binary class datasets by considering the class with fewer samples as minority class and rest of the samples as majority class as suggested by Wu and Edward [139]. From Table 6.1, C and γ refer to the parameters of the SVM classifier such as loss function and RBF kernel width.

Table 6.1: Datasets Description

Dataset	#Min	#Maj	Imbalance ratio	#Attributes	Class(Min, Maj)	SVM parameters	
						C	γ
Glass	17	197	11.58	8	(Ve-win-float-proc, remainder)	100	10
Ionosphere	126	224	1.77	34	(b, g)	1	0.2
Pima	268	500	1.87	8	(1, 0)	5	0.2
Waveform	1647	3353	2.035	21	(1,remainder)	1	default in LIBSVM
Letter-a	789	19211	24.4	16	(a, remainder)	1	default in LIBSVM
Satimage	626	5809	9.27	36	(4, remainder)	50	0.2
Abalone	102	4075	39.9	8	(15,remainder)	100	0.5
Gamma	6688	12332	1.84	10	(h, g)	1	default in LIBSVM
Shuttle	8903	49097	5.51	9	(4, remainder)	50	2

6.4.2 Discussion on StatQSVM

Experimental evaluation is carried out over 5 individual training and test sets of each dataset (5 fold cross validation) and average $F-measure$ is considered as the performance evaluation criterion. Training set is obtained by sampling 90% of data from entire dataset, and test set is obtained by sampling rest of the 10% of the dataset. The instances taken as test set are different from those in training set. LOB [39] and StatQSVM [86] algorithms were re-implemented in MATLAB environment using LIBSVM [15] as a background tool for support vector machines. Our first concern is to analyze the effect of class imbalance on probabilistically queried active learning criterion. In order to do thorough analysis we have considered nine unbalanced datasets with different imbalance ratios ranging from 1.84 to 39.9 and with sizes of the datasets ranging from 200 to 58,000 (see Table 6.1). For comparing the behavior of LOB and StatQSVM active learning algorithms in class imbalance settings, results on entire training set are considered rather than applying early stopping condition. Fig. 6.2 depicts the behavior of active learning algorithms in various imbalance settings. We observed that for moderate unbalanced datasets like *Pima*, *Glass*, *Waveform*, *Letter_a* and *Gamma*, $F-measure$ stabilizes after few iterations of StatQSVM algorithm. This indicates that StatQSVM algorithm is also capable of querying minority samples at the early rounds of its execution. Moreover, for *Pima*, *Gamma*, *Waveform* and *Glass* datasets, StatQSVM converged to the solution earlier than LOB method. For Glass and Waveform datasets StatQSVM attained superior $F-measures$ 0.66 and 0.86 respectively with early stopping condition. For Letter-a and Shuttle datasets the performance of the StatQSVM is similar to LOB after few iterations of querying process. But for *Satimage* dataset, StatQSVM algorithm did not exhibit consistent performance in terms of *F-measure*. This might be due to the non-separable nature of the data. For highly unbalanced *Abalone*, datasets, StatQSVM and LOB, do not yield significant $F-measure$. The graph of *Abalone* dataset reveals that for both algorithms after few iterations adding more points to the learned model leads to drop in performance towards zero. In this case the performance of StatQSVM algorithm drops earlier than the LOB method. This experimental study discloses the fact that StatQSVM is also efficient as QuerySVM algorithm in handling moderate unbalanced datasets. But from Abalone data set, it is clear that both algorithms could not cope with highly unbalanced datasets in

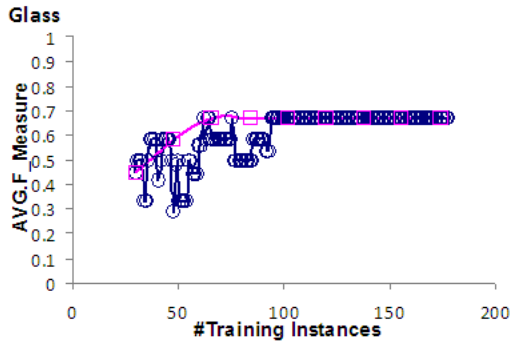
yielding better $F-measure$.

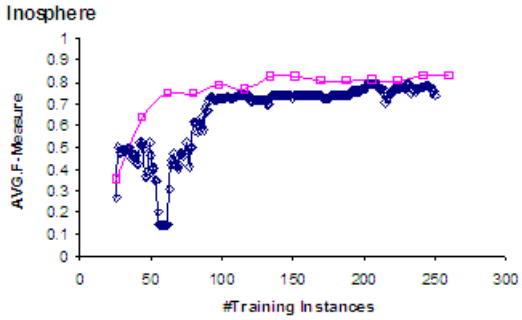
6.5 Proposed Algorithm

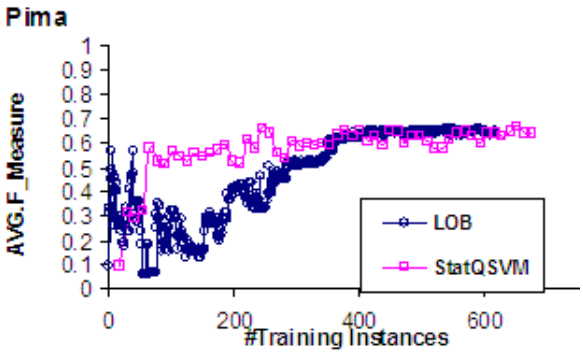
From the exploration of StatQSVM (see Sec 6.4) it is observed that for highly unbalanced datasets the performance improvement in terms of minority class *F-measure* is not significant and it may drop down after adding few samples to the current model. Experiments on *Satimage* and *Abalone* datasets substantiate this observation. The research of Wu and Edward [139] concluded that more the class imbalance in the dataset more skewed is the boundary towards minority class in SVM based learning methods resulting in the lose of generalization for minority class. Consequently, in case of active learning with SVM, at initial iterations, if the boundary is skewed towards the minority class, then adding more number of samples to these learned model leads to gradual performance drop-down. To address this problem, this chapter proposes to learn cost-weighted hyperplane with the loss functions C^+ and C^- set to imbalance ratio of $S = S \cup q$ in every iteration of probabilistic active learning. The proposed algortihm is shown in Algorithm 9.

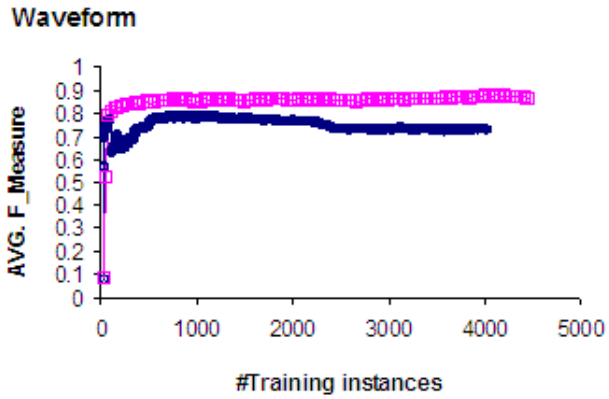
6.5.1 CStatQSVM Algorithm

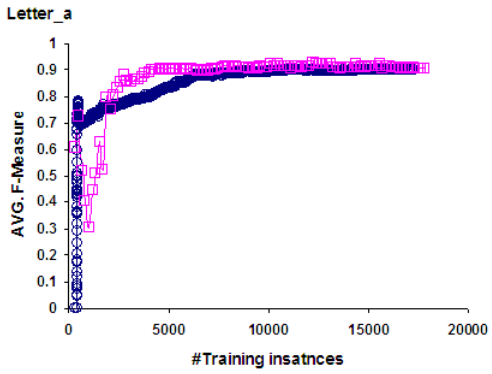
Let us suppose that, $D = \{x_1, x_2, x_3....x_n\}$ is the whole training set used for SVM model construction through active learning. Let P_t is the training set at iteration t with random points from D, i.e., $P_t \subset D$ and $q = size(P_t)$. Assume that $S_t = SV(P_t)$ is a set of support vectors $\{s_1, s_2, s_3, ...s_l\}$ obtained using the cost-weighted methodology discussed in section 6.3.1 and $\langle w_t, b_t \rangle$ be the corresponding separating hyperplane during the iteration t. The loss functions C^+ and C^- for $\langle w_t, b_t \rangle$ are set to the imbalance ratio $IR_t = (S_t \cup P_t)$, where q_t are q random points that are actively queried at iteration t. In each iteration, the degree of confidence factor c which estimates the likelihood of each point in q that it belongs to S is calculated using eq. 6.5. The algorithm for CStatQSVM is given in 9. According to Morik et al. [87] assigning C^+ and C^- to IR_t avoids boundary skew through balancing the precision and recall trade-off by increasing true positive rate and decreasing true negative rate. Therefore, the cost-weight based probabilistic active learning (CStatQSVM) enables querying minority class samples at the initial iterations of active learning process, thus avoids performance drop in the prediction.

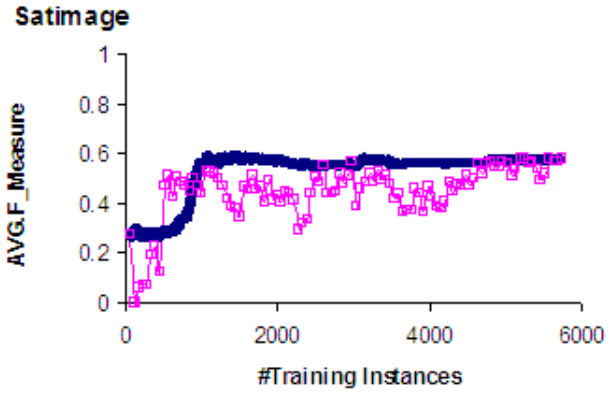

Glass
AVG.F_Measure
#Training Instances

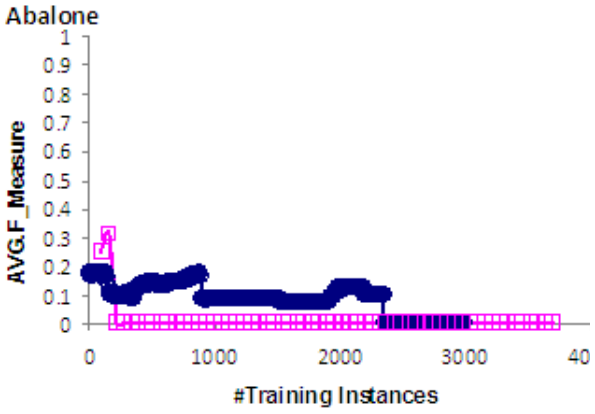

Inosphere
AVG.F-Measure
#Training Instances

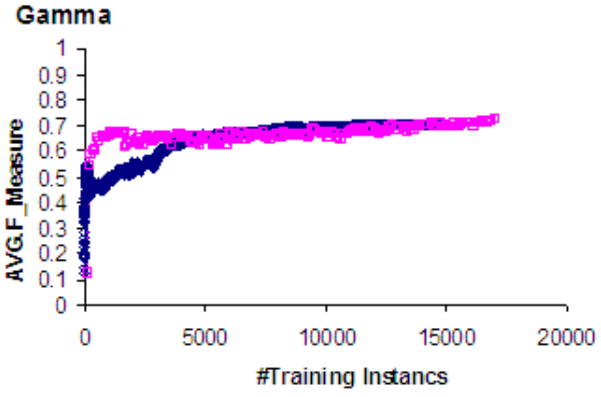

Pima
AVG.F_Measure
LOB
StatQSVM
#Training Instances


Waveform
AVG. F_Measure
#Training instances


Letter_a
AVG. F-Measure
#Training insatnces


Satimage
AVG.F_Measure
#Training Instances


Abalone
AVG.F_Measure
#Training Instances


Gamma
AVG.F_Measure
#Training Instancs

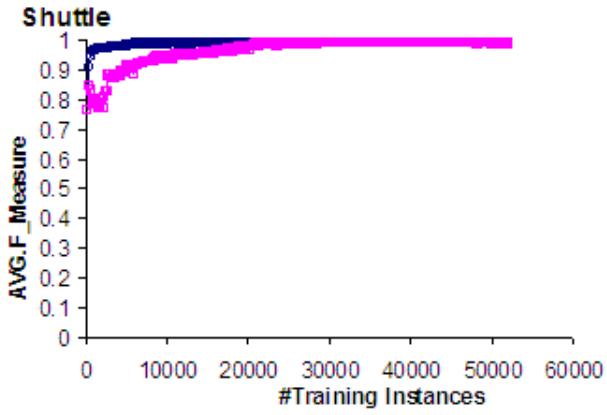


Figure 6.2: Comparison across StatQSVM and LOB active learning methods in terms of *F-measure*

6.5.2 Stopping Criterion

Ideally SVM active learning can terminate when there are no more informative instances left in training set to improve the SVM margin in terms of classification performance, i.e., the margin is exhausted.

To take advantage of convergence to maximum performance, a stopping criterion has to be device for CStatQSVM in case of unbalanced datasets. This chapter proposes a stopping criterion for CStatQSVM based on stabilization of confidence factor c.

Previously, Ertekin et al. [39] suggested stopping criterion for LOB approach, which is used when the count of support vectors is stabilized. In this approach the obtained final support vectors (SVs) through active learning reflect the actual margin of SVM classifier. Once all informative instances in training set are queried by LOB, no change in support vector count was observed. The authors considered initial stabilized point as a stopping criterion for active learning.

Unlike the LOB method, the support vector set S at iteration t in CStatQSVM represents the relative closeness to the actual support vectors. The confidence factor c reflects the degree of closeness from current boundary to the actual boundary. Therefore, once current support vector set S_t reaches the actual support vector set, adding more number of samples to this learned model cannot improve the confidence factor significantly. This behavior was analyzed on three individual random test sets of *Gamma* and *Satimage* dataset with two different kernel settings. The

Gamma dataset learns with linear kernel and *Satimage* learns with RBF kernel. Constructed test sets are actual representation for training sets. From Fig. 6.3 the analysis on *Gamma* and *Satimage* datasets reveals that the degree of confidence stabilizes after few iterations of querying process for both StatQSVM and CStatQSVM i.e no remarkable change is observed in the confidence factor even after adding newly queried samples to the current SVM model. Furthermore, for CStatQSVM the confidence factor c saturates earlier than StatQSVM. This might be due to setting the loss functions C^+ and C^- to imbalance ratio of $S = S \cup q$ during every iteration of CStatQSVM which leads to query in almost all minority samples at early rounds of execution itself. Therefore, the degree of confidence for CStatQSVM stabilizes when all the minority samples are examined in the training set. This indicates that the margin is exhausted for CStatQSVM active learning. Furthermore, it is also identified that once the margin is exhausted there is no significant improvement in minority class $F - measure$. Average results on 3 test sets, confidence factor c and $F - measure$ are shown in Fig.6.3. The vertical line in Fig. 6.3 represents the stopping point for CStatQSVM active learning. This stopping point is determined on confidence factor c of CStatQSVM as the difference among four consecutive runs and the algorithm stopped if the value is less than a threshold 0.01.

6.6 Empirical Evaluation of CStatQSVM

This section presents the evaluation of proposed CStatQSVM and discuses the results. Furthermore, this section also provides a comparative study with different active learning methods and the other methods that address class imbalance issue to study the relative performance of the proposed approach.

The evaluation of CStatQSVM was carried out over several datasets with varied degrees of imbalance that are considered in Section 6.4. Moreover, CStatQSVM is implemented in MATLAB environment and LIBSVM's cost-weighted parameter w_i is used for tuning C^+ and C^- of CStatQSVM.

6.6.1 Discussion on CStatQSVM

The triangle symbol line in Fig.6.4 depicts the behavior of CStatQSVM on different unbalanced datasets. Except *Letter-a* dataset CStatQSVM exhibited consistent

Algorithm 9 Pseudocode for CStatQSVM algorithm

Initialize: Let P_0=Randomly selected initial set from training set D;

$q = size(P_0)$;

$t = 0$;

$S_0 = SV(P_0)$; /* Support vectors obtained from training set P_0* \

$IR_0 = \frac{\#Majority\ Class\ samples\ in\ P_0}{\#Minority\ Class\ samples\ in\ P_0}$;

Let the parameters of the current hyperplane be$\langle w_0, b_0 \rangle$;

$D = D \setminus P_0$;

Begin

while *StoppingCriterion* is not satisfied **do**

$P_t = \phi$

while $size(P_t) \leq q$ **do**

Randomly select an instance $x \epsilon D$. Let y be the label of x and $\langle w_t, b_t \rangle$ be the parameters of current hyper plane

if $y(w_t.x + b) \leq 1$ **then**

Select x with probability c. Set $P_t = P_t \cup x$;

else

Select x with probability $1 - c$. Set $P_t = P_t \cup x$;

end if

end while

$S_t = SV(S_t) \cup P_t$;

$P_t = S_t$;

$D = D \setminus P_t$;

$IR_t = \frac{\#Majority\ Class\ samples\ in\ P_t}{\#Minority\ Class\ samples\ in\ P_t}$;

Train SVM on P_t with misclassification costs C^+, C^- which are assigned with IR_t (discussed in section 6.3.1).

t=t+1;

end while

End

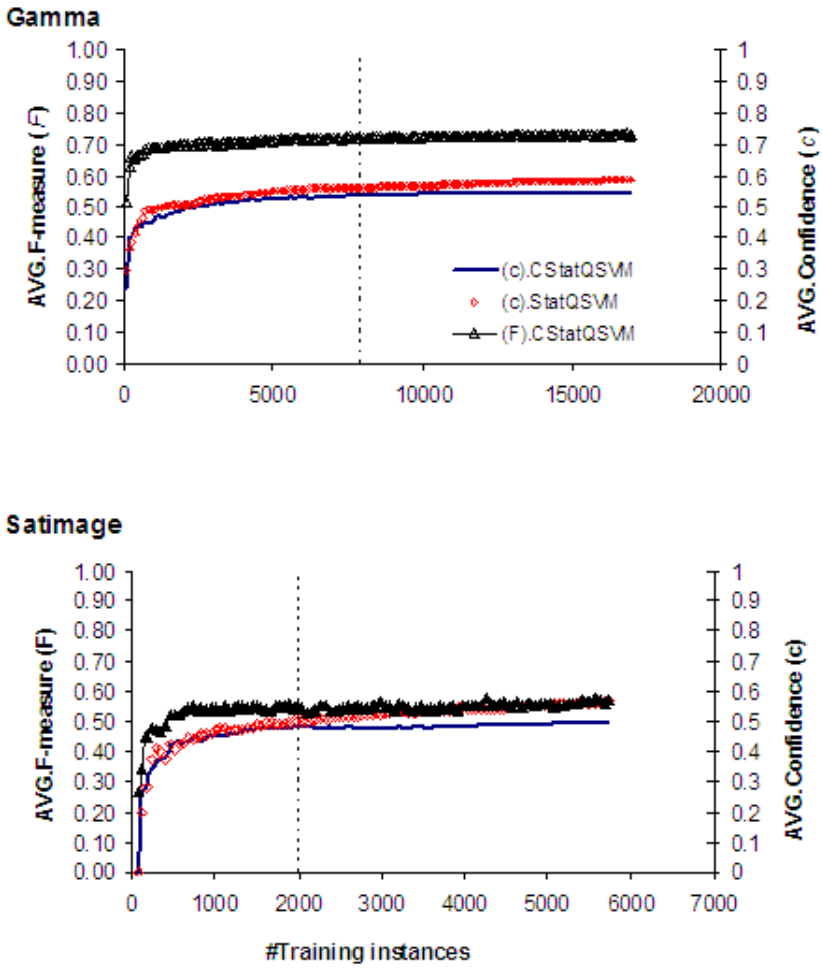


Figure 6.3: Stopping criterion for CStatQSVM algorithm. Vertical line indicates the stopping point

performance in terms of $F-measure$ till the end of training set. Similar kind of behavior is identified for highly unbalanced dataset *Abalone*. As shown in Table 6.2, for this dataset CStatQSVM algorithm exhibited 40% improvement in $F-measure$ compared to StatQSVM and LOB algorithms. Over all datasets compared to StatQSVM and LOB, CStatQSVM converges faster to the solution.

The CStatQSVM undersampling algorithm was compared with other solutions of class imbalance problem:

- Random undersampling (RUS).
- Cost weighted hyperplane (DEC) of whole dataset.

Since class imbalance affects the SVM boundary obtained on whole dataset, a comparison with Batch SVM, was also provided to constitute a baseline for all experiments. The vertical line at right middle of each diagram in Fig.6.4 represents the early stopping criterion for the CStatQSVM algorithm. The average $F-measure$ results listed in Table 6.2 are with the adoption of early stopping criterion on CStatQSVM. The stopping point for each dataset was determined as mentioned in section 6.5.2. To compare the performance of the active learning algorithms, the same stopping point of CStatQSVM was set to StatQSVM and LOB. There is no early stopping criterion for the rest of the methods considered for the comparison due to their passive learning nature. $F-measure$ is used as an evaluation metric for estimating the minority class prediction across all methods considered for experimental study.

From Table 6.2, it is observed that Batch SVM classifier yields zero minority class $F-measure$, which indicates that for highly unbalanced non-separable datasets such as Glass and Abalone. Moreover, for *Abalone* dataset in active learning setting, it is observed that as more number of samples is added to the boundary the performance dropped towards zero. Fig.6.4 depicts this scenario (see *Abalone*). However, for this dataset CStatQSVM algorithm exhibited consistent performance throughout the active learning process. Furthermore, for *Glass* dataset the active learning algorithms significantly improved the minority class $F-measure$ from 0 to 66%. Therefore, from the above described evidence, we conclude that undersampling through active selection of samples is a good aid for improving classifier performance in extreme imbalance cases.

As for random undersampling (RUS) from Table 6.2, it balances the class distribution by randomly selecting equal number of majority class samples from the

Table 6.2: Comparison of Average $F-measure$ of minority class and training time across the methods

Dataset	*F-measure*						AVG.Time(Sec)			n_{Select}
	Batch	RUS	DEC	LOB[39]	Stat QSVM[86]	CStat QSVM	LOB[39]	Stat QSVM[86]	CStat QSVM	
Glass	0	0.333	0	**0.666**	**0.666**	**0.666**	0.71	**0.14**	*0.17*	78
Ionosphere	0.808	**0.828**	0.801	0.738	0.801	0.808	2.5	**0.410**	*0.415*	170
Pima	0.624	0.612	0.51	0.59	0.63	**0.651**	21.1	**2.6**	*3*	379
Waveform	0.85	0.84	0.82	0.78	**0.86**	**0.86**	220	**179**	*180*	1890
Letter-a	0.9	0.71	0.7	0.9	**0.92**	0.88	360	**84**	*170*	7,050
Satimage	0.547	0.259	0.12	**0.565**	0.404	0.55	338	**80.3**	*125.6*	2000
Abalone	0	0.317	0.06	0.15	0	**0.48**	317.45	**21.47**	*52.72*	1540
Gamma	0.7	0.62	0.55	0.66	0.66	**0.71**	562	**369**	439	8100
Shuttle	**0.996**	0.862	0.966	0.994	0.993	0.972	1792.9	**699.15**	*1265.23*	26300

majority class distribution as there in the minority class. For large datasets such as *Shuttle*, *Letter-a*, *Gamma* and *Satimage* datasets, the performance of the random undersampling (see Table 6.2) is significantly less compared to Batch and remaining active learning methods. This degradation in performance is due to the loss of informative instances from majority class in order to form appropriate SVM boundary. Except the small *Ionosphere* dataset active learning methods exhibit better performance than random undersampling. The performance improvement is due to an intelligent search that is carried out on whole dataset through active learning which enables selection of the informative samples from both classes. From Table 6.2, n_{select} represents the number of informative samples selected through CstatQSVM with stopping criterion discussed in Section 6.5.2. This indicates that considerably less number of actual informative instances required for training CstatQSVM from entire training set.

In order to facilitate the analysis of results obtained during the comparative study, obtained $F-measures$ are ranked across different methods using Friedman's ranking method as described in [37] (see Table 6.3). Among all methods proposed CStatQSVM received best mean rank of 4.88 on minority class $F-measure$. StatQSVM stood in second place. DEC stood in last position. The query based active learning method proposed for handling unbalance datasets LOB, stood in third position. From this ranking evidence, it is concluded that the proposed CStatQSVM is significantly accurate compared to all the other methods considered for comparison purpose. Further Wilcoxon signed rank test showed that CStatQSVM can be statistically better than random undersampling and DEC on zero median hypothesis which is rejected at 5% significant level of p=0.0078 and 0.0039 respectively. Last three columns in Table 6.2 depict the comparative study of the computational efficiency across active learning methods. The StatQSVM active learning is faster than CStatQSVM and LOB methods in terms of computational time. But the computational complexity of CStatQSVM is superior to LOB method. Compared to StatQSVM, the degradation of computational efficiency in CStatQSVM is due to the extra computational overhead caused by the loss function calculation of C^+ and C^-. However from classification point of view CStatQSVM with stopping criterion is superior to all other methods. Therefore, it is concluded that CStatQSVM based undersampling is an efficient algorithm in handling class imbalance problem with respect to both classification performance

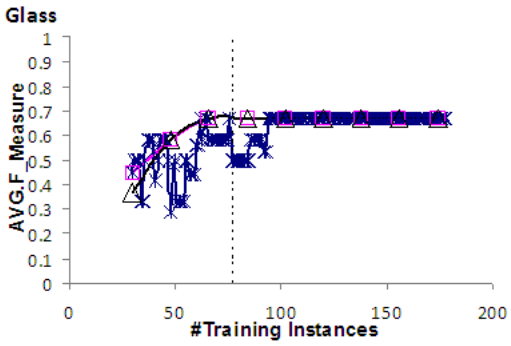
Table 6.3: Ranking of minority class *F-measure* across different methods

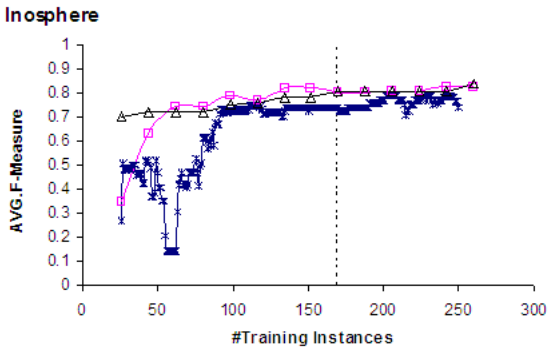
Dataset	Ranking across different Methods					
	Batch	RUS	DEC	LOB[39]	StatQSVM [86]	CStatQSVM
Glass	1.5	3	1.5	**5**	**5**	**5**
Ionosphere	4.5	**6**	2.5	1	2.5	4.5
Pima	4	3	1	2	5	**6**
Waveform	4	3	2	1	**5.5**	**5.5**
Letter-a	4.5	2	1	4.5	**6**	3
Satimage	4	2	1	**6**	3	5
Abalone	1.5	5	3	4	1.5	**6**
Gamma	5	2	1	3.57	3.5	**6**
Shuttle	**6**	1	2	5	4	3
AVG.Rank	**3.88**	**3**	**1.66**	**3.55**	**4**	**4.88**

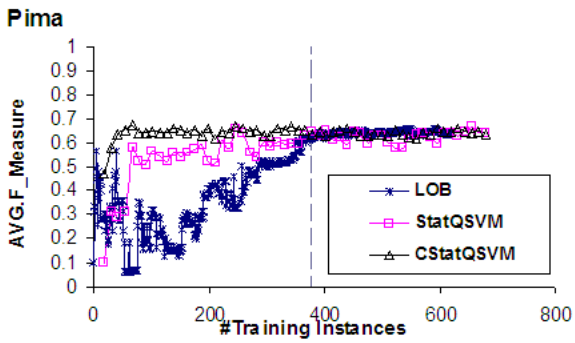
and computational complexity compared with other active learning and the other passive solutions for handling class imbalance problem.

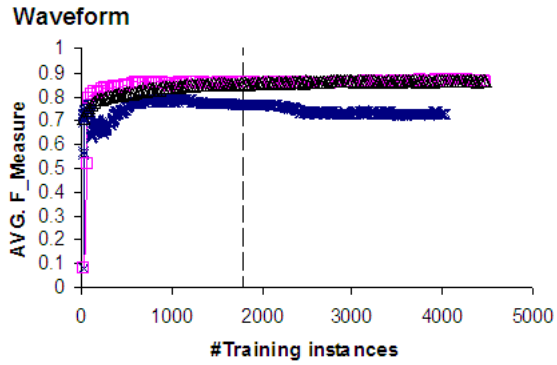
6.7 Chapter Summary

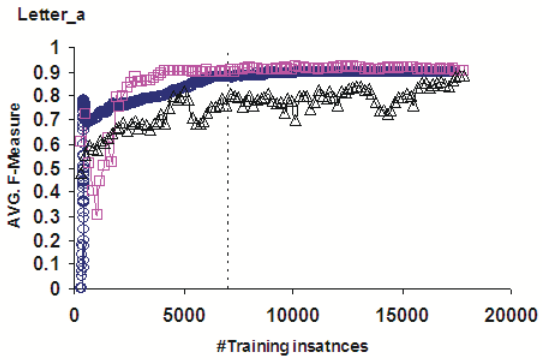
This chapter explores the viability of probabilistic SVM active learning (StatQSVM) for the challenging scenario of class imbalance problem. The experimental evidence indicates that StatQSVM is effective than query based active learning method (LOB) in terms of performance and computational efficiency. Moreover, for few datasets StatQSVM method converges to balanced class distributions earlier than the query based active learning methods. In order to improve the performance of StatQSVM, for highly unbalanced non-separable datasets, we have proposed a cost-weighted probabilistic undersampling algorithm with a new early stopping criterion called CStatQSVM. Obtained experimental results point out that the proposed algorithm yields better performance than LOB method within significantly lesser time. The comparative study with other passive learning methods for class imbalance problem revealed that proposed CStatQSVM attained superior performance. Although the proposed method takes more time compared to StatQSVM, it has superior minority class classification performance.

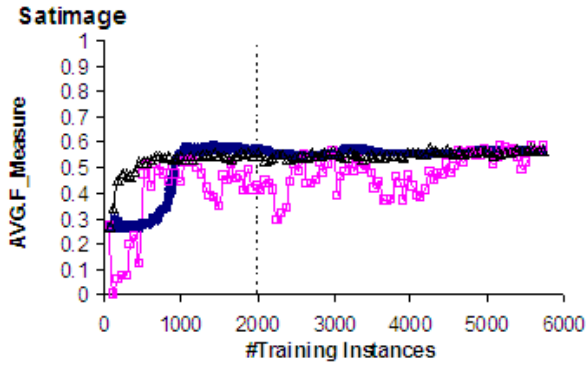

Glass
AVG.F_Measure
#Training Instances

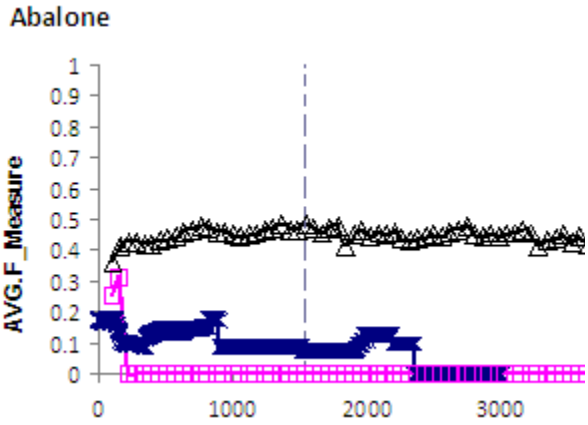

Inosphere
AVG.F-Measure
#Training Instances

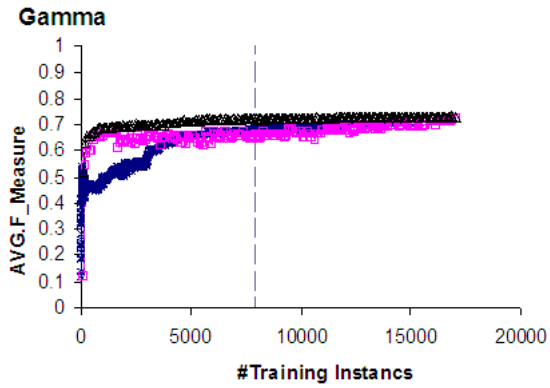

Pima
AVG.F_Measure
#Training Instances
LOB
StatQSVM
CStatQSVM


Waveform
AVG. F_Measure
#Training instances


Letter_a
AVG. F-Measure
#Training insatnces


Satimage
AVG.F_Measure
#Training Instances


Abalone
AVG.F_Measure


Gamma
AVG.F_Measure
#Training Instancs

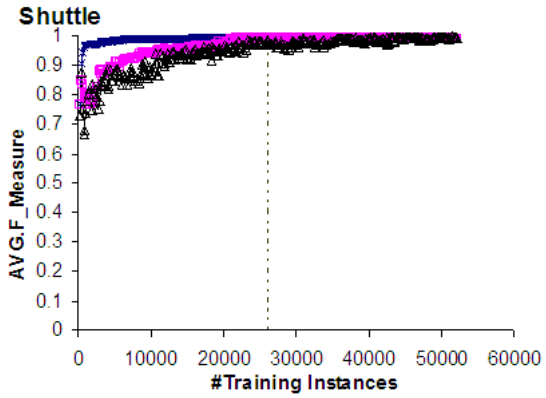


Figure 6.4: Comparison of CStatQSVM performance with StatQSVM and LOB in terms of *F-Measure* with early stopping criterion.The vertical line indicates the early stoping point.

Preprocessing + Informative oversampling

Chapter 7

Is PCA Effective for Preprocessing Unbalanced Data?

As discussed in chapter 2 several classifiers seem to get biased towards majority class in case of unbalanced datasets. This chapter investigates whether the dimensionality reduction using Principal Component Analysis (PCA) for two-class classification problems is affected by unbalanced distributions. The rest of this chapter is organized as follows. Section 7.2 presents the related work. Section 7.3 presents the basic steps for Principal Component Analysis and elaborates its possible effects due to unbalanced data. Section 7.4 depicts the evaluation measure for reconstructing original unbalanced data space. The experimental study over series of two-dimensional synthetic datasets and real-world datasets from UCI repository [9] that reflect the effect of unbalanced datasets on PCA are described in section 7.5 and 7.6, respectively. This chapter is summarized in section 7.7.

7.1 Introduction

For high-dimensional data, classification process may also include dimensionality reduction to increase the class discrimination, better data representation and for attaining good computational efficiency. The misclassification rate for classification process drastically increases due to spurious dimensions in the original high dimensional data space, known as curse of dimensionality problem [36]. Therefore, dimensionality reduction is required. There are two commonly used approaches to reduce the dimensions: supervised (Linear Discriminant Analysis) and unsu-

pervised (Principal Component Analysis). It is an agreed fact that Principal Component subspace (PC subspace) is adequate to hold the discriminative information for classification problem. PCA linearly transforms high dimensional data into lower dimensional space by maximizing the global variance of the data as well as minimizing the least square error for that transformation. But from Vaswani et al. [122] for classification problems with unequal size covariance structure (one class covariance structure is different from other class) PCA can not discover class discriminative information. In the case of unbalanced datasets, the spread is dominated by majority class as its prior probabilities are much higher than minority class samples. Capturing and validating labeled samples, particularly non-majority class samples, are very costly and challenging. In this chapter, we apply Principal Component Analysis (PCA), which is an unsupervised method for dimensionality reduction. This work focuses on effect of PCA in terms of directional difference between principle axes over unbalanced datasets.

In this chapter, we present an explorative study on whether unbalanced datasets affect the subspace generated by unsupervised dimensionality reduction method like PCA while reducing the original high dimensional space for two-class classification problems. We have considered both balanced and unbalanced data for the investigation. As a part of this contribution, experiments are conducted both on synthetic and real world unbalanced datasets with three classifiers namely DT, NB and kNN. The experimental results conclude the following:

- Whenever the directions of principal axes of both classes are different, the reduced subspace by PCA on unbalanced data favors the majority class subspace. This is mainly due to the dominating nature of maximum variance directions of the PCs by the majority class.

- Whenever the directions of both majority and minority classes fall on the same principal axis, the effect of unbalanced datasets on PC subspace and classification performance is not prominent as the PCs found are equally good for both majority and minority classes.

7.2 Related Work

Several researches studied the behavior of various classifiers on different characteristics of unbalanced data. Most of the researchers concluded that the hardness

of the class imbalance problem is not only due to class discrimination property of the classification algorithm used but also due to the internal data characteristics. Xie and Qie [141] have proved that Linear discriminant analysis (LDA) on Gaussian distribution assumption biases towards the majority class. They have concluded that the unequal size covariance matrices are the key reason to behave so. Their studies have also shown that by balancing the original unbalanced distribution the performance of the LDA algorithm can be improved. However, Hao and Titterington [52] have disproved the earlier claim on LDA. They have shown experimentally that the unequal covariance matrix is not a key reason for biasing LDA towards the majority class. They also proved that the resampling over original unbalanced distribution causes negative effect on LDA. Mazurowski et al. [84] trained classical back propagation neural networks and particle swarm optimization (PSO) on clinically relevant training data. The authors concluded that while training neural networks along with class distribution, imbalance ratio and training sample size, the number of features being considered also play a key role in factor for performance degradation.

Principal Component Analysis (PCA) is a widely used dimensionality reduction technique in high dimensional multivariate classification problems like face recognition, object recognition and handprint recognition [83]. Chandana et al. [28] have extracted PCA features from the balanced datasets obtained by balancing techniques such as SMOTE and random undersampling. The combined technique of undersampling and SMOTE has been used for predicting the stage of prostate cancer. Apart from the dimensionality reduction technique, PCA can be used as a classifier for null space analysis. For example Vaswani et al. [122] have proposed a classifier for object recognition problem where different classes have unequal covariance matrices (the major directions of one class are different from other class). In the first step of the approach, they project the whole dataset into the PCA subspace to retain the maximum variance directions between the classes. After that the algorithm starts to find M_i trailing eigen vectors for each class, i, along the directions where the intra-class variance is smallest (called null space). The choice of M_i depends upon other factors of the assumptions of the algorithm. Finally, during the testing step an unseen sample is first projected into PCA space and later classified over the corresponding null space of each class.

The experimental studies presented in this chapter reveal that for a global di-

mensionality reduction technique (e.g. PCA), presence of class imbalance can lead to discriminative information loss and degradation in classification performance from the under represented minority class. This is very prominent in case of higher directional difference between the principal axes of two classes.

7.3 PCA and its Possible Effects

One of the main strengths of PCA is to reduce the complex data space into lower dimensional space without losing the global structure of the data. PCA has large scale of applications, because of its simplicity and non-parametric nature. The basic assumptions of PCA are as follows [66].

- The mean and variance is sufficient statistics to represent the probability distributions of the whole dataset.
- The dataset is linear in nature, with a combination of certain basis vectors.
- The maximum variance directions in the data are associated with important dynamics, whereas the lower variance directions are associated with noise. The better directions may not contribute relevant information for improving accuracy of the classification process.

The PCA algorithm using eigen vector implementation [66], is described below.

1. Let $X = \{x_1, x_2, x_3, ...x_N\}$ be a set of M dimensional sample vectors with N number of samples.
2. Compute the mean (μ) of the whole data and use it to center the data towards the mean by $\widehat{x_i} = x_i - \mu$. Here $i = 1, 2, 3, .., N$; $\widehat{x_i}$ is the shifted sample.
3. Compute the total covariance matrix $C_t = \widehat{x_i}\widehat{x_i}^T$ of size $M \times M$ from all the samples.
4. Find transformation matrix to maximize the total covariance matrix.

$$W_{PCA} = argmax|\widehat{C}_t| \tag{7.1}$$

where $\widehat{C}_t = WC_tW^T$.

The above equation helps us to obtain the main objective of PCA (transform

the data to a subspace where the maximum variance directions of the original data are retained). Eq. 7.1 is an optimization problem with eigen value solution. By solving the eigen value equation, the columns of the matrix W constitute M eigen vectors of M length from the covariance matrix C_t.

5. Arrange W_j eigen vectors of W in the order of decreasing magnitude of the corresponding eigen values λ_j, that is, sort the eigen vectors in order of decreasing magnitude of corresponding eigen values.

6. Let $\widehat{W}_{PCA}$ is new transformed matrix obtained by discarding $(M-R)$ eigen vectors from W_{PCA} . Here, R is required reduced dimensions.

7. The final R sub dimensional projected matrix is $\widehat{Y} = \widehat{W}_{PCA}X$.

8. The original data can be reconstructed as $X^{'} = \widehat{W}_{PCA}\widehat{Y}$, with approximated Mean Square Error $\sum_{k=R+1}^{M} \lambda_k$.

9. The $RMSE$ for the new data transformation is calculated as $\sqrt{(X - X')^2}$

According to geometrical properties of PCA [66] for a dataset X of p-dimensions, if $XC_tX = \text{constant}$, where C_t represents the covariance of the population and the PCs define the principal axes of these p-dimensions. Motivated by this reasoning, we have modeled our synthetic study in the form of principal axes of the datasets, which are inherently the principal components for PCA algorithm.

Usually, the global mean in step 2 represents the center of the whole data. However, in case of unbalanced datasets, the global mean is moved towards the content of the majority class. This is because of the dominating nature of prior probability of majority class compared to that of minority class. Furthermore, the covariance matrix, mentioned in step 3, captures the spread of the dataset. If the data is highly skewed (highly unbalanced), the spread of the majority class itself predominate while computing the covariance matrix. Moreover, the principal axes directions with maximum variance from the whole data may be different from those of minority class samples. Therefore, recovery of the minority class data from principal components obtained on unbalanced data and prediction of minority class samples over the reduced subspace is an important issue. In order to study the impact of class imbalance over principal axes, we conducted series of experiments as explained in the next sections.

Though $RMSE$ is a measure for computing the goodness of transformation to the lower dimension, computing $RMSE$ from the total dataset may not give clear picture of minority class recovery. Therefore, there is a need for computing $RMSE$ separately for majority class and minority class data points.

7.4 Evaluating PCA Performance over Unbalanced Datasets

The performance of the dimensionality reduction techniques is measured with how well the original data can be reconstructed from the reduced dimensions. This can be achieved through Root Mean Square Error ($RMSE$) between the original data and reconstructed data using reduced dimensions. This is also called *reconstruction error* for constructing original data from reduced dimensions. As the number of principal components increases, the $RMSE$ gradually reduces and may fall to zero at full dimension. Commonly, maximum variance principal axes are used for finding the optimal number of dimensions for a transformation in classification problems. $RMSE$ also represents maximum variance principal axes in some way. Sum of squared perpendicular distances is one such similar measure of goodness-of-fit [66]. The definition of $RMSE$ for the PCA transformation is described below:

$$RMSE = \sqrt{(X - X')^2} \tag{7.2}$$

In eq. 7.2, X represents a data point in original unbalanced input space and $X^{'}$ is the same data point represented in transformed PC subspace.

In real world unbalanced data classification scenarios like fraud detection, the cost associated with wrong prediction of rarely occurring (fraud) samples is very high and predicting these samples is always valuable for the organizations [95]. Therefore, the minority class subspace generated by PCA should be better representative of the original minority class data for unbalanced classification problem. The present work studies this property by computing average reconstruction error from the individual minority and majority classes separately.

7.5 Experiments on Synthetic Datasets

This section presents two experiments that are carried out on synthetic datasets by manipulating the angle of separation between the principal axes corresponding to majority and minority classes. Experiment-A with the directions of the two principal axes are different and Experiment-B - with the same principal axis direction, for majority class and minority class data. The experiments are carried out on the datasets with different imbalance ratios. Since the characteristics of the artificially generated datasets are more accessible than the real-world datasets, we analyze two scenarios, 5000 samples for each dataset have been generated for the study. Each dataset constitutes both minority and majority class with varying imbalance percentages ($1\%, 5\%, 10\%, 20\%, 30\%, 40\%$ *and* 50% from minority class, and the remaining samples from the majority class).

To study the effect of imbalance datasets on reduced PC subspace, we applied PCA on artificially generated two dimensional datasets and reduced the input space to one dimensional subspace. Later PCA performance over unbalanced data is evaluated using the measure as described in section 7.4. Within the evaluation, we have calculated the average reconstruction error of two classes separately. Furthermore, we explored the relation between reprojection capability and the performance of the classifier over reduced dimensions, by varying imbalance class distributions. Further, the classification performance is compared with the classifier learned on the original unbalanced datasets. kNN classifier is used for these experiments because of its simplicity and local learning.

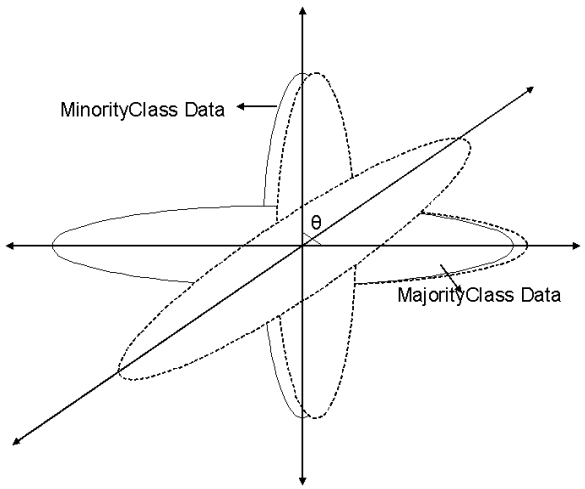


Figure 7.1: Schematic diagram for the sample generation in angular separation.

7.5.1 Experiment-A

The first experiment is aimed at analyzing the behavior of PCA in the case of class imbalance problem when the majority and minority classes fall in different maximum variance principal axis directions. In order to simulate this behavior, artificial samples are generated with angular separation (θ) between the majority class and minority class principal axes, where $\theta = 90^0,\ 60^0,\ 30^0\ and\ 0^0$. For each of these cases, 7 datasets with different imbalance ratios staring from 1% to 50% were generated. Each dataset contains majority and minority class data centered at (0, 0) with variances 15 and 2 along the axes.

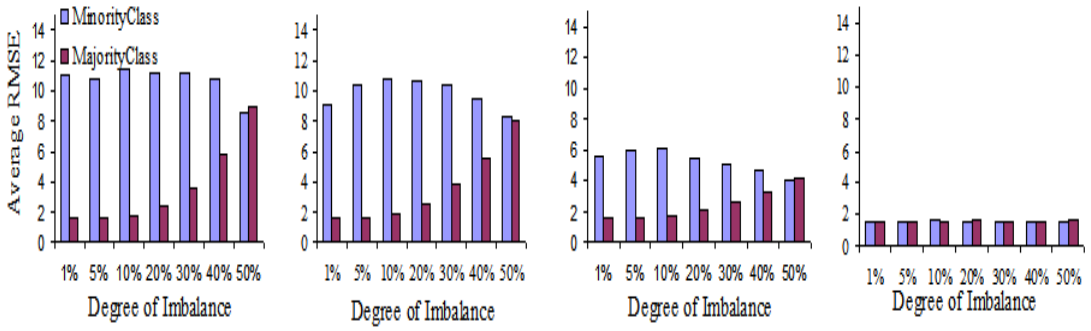


Figure 7.2: Reprojection error for $90^0,\ 60^0,\ 30^0\ and\ 0^0$ angular separations with varied Degree of Imbalance ratios.

Fig 7.1 depicts the schematic diagram for generation of synthetic samples in different principal axes directions for majority and minority classes in a two dimensional dataset. The solid ellipsoids in Fig 7.1 represent initial positions (angular separation 90^0) of the two classes and the dotted ellipsoids represent the angular rotation of the majority class towards the minority class data ($60^0,\ 30^0\ and\ 0^0$).

Figures 7.2, 7.3 and 7.4 present bar graphs of RMSE for various angular separations ($90^0,\ 60^0,\ 30^0\ and\ 0^0$) as well as for different imbalance ratios (1%, 5%, 10%, 20%, 30%, 40%, *and* 50%). The x-axis of these figures represents different degrees of imbalance between majority and minority data. Fig 7.2 shows the reprojection error (average $RMSE$) for each class. Fig 7.3 and Fig 7.4 describes kNN classifier performance in terms of $F - measure$ computed both for majority and minority classes. Here, Fig 7.3 depicts the performance on original data and Fig 7.4 repre-

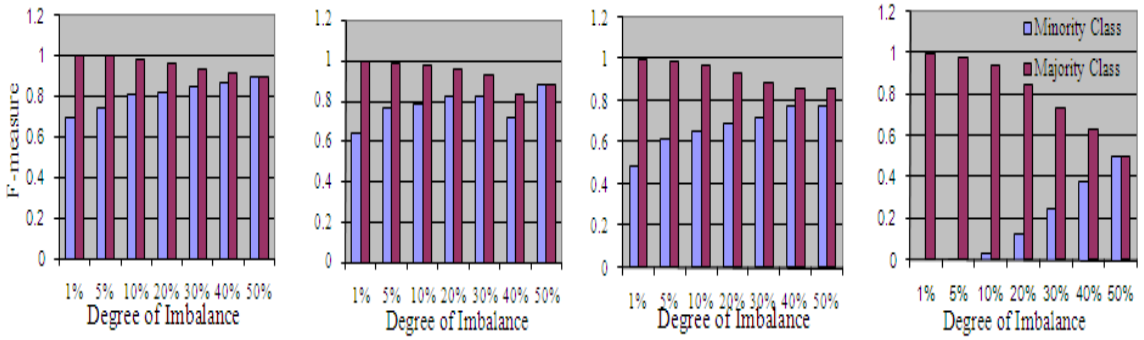


Figure 7.3: $F-measure$s from kNN classifier for minority and majority class on original data for $90^0,\ 60^0,\ 30^0$ *and* 0^0 angular separations.

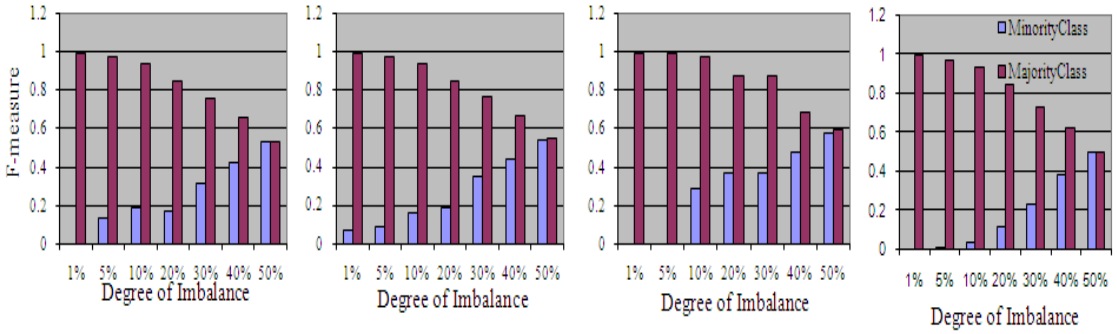


Figure 7.4: $F-measure$ from kNN classifier for minority and majority class on reduced dimensions of PCA for $90^0,\ 60^0,\ 30^0$ *and* 0^0 angular separations scenario of PC's.

sents performance on first principal component.

The main intuition behind these experiments is that usually for unbalanced datasets the minority samples are under-represented (low in number) than the majority samples. When the directions of principal axes for both classes are different, the maximum variance contributed by the minority class principal axes are comparatively less than that of majority class. According to PCA, the maximum variance directions represent important dynamics of the data, and thus the principal components obtained on imbalance datasets are dominated by the majority class maximum variance directions. Therefore, the principal components obtained on class imbalance datasets with different principal axes from two classes can lose the discriminative information of the under represented minority class samples by resulting in huge reprojection errors from the minority class. To prove this assertion for each angular separation scenario, different imbalance ratios starting from 1:99 to 50:50 are considered for the experimental study.

From the reprojection errors and kNN classifier performance of Figures 7.2, 7.3 and 7.4, we observed the following:

1. Except the 0^0 angular separation (see Fig 7.2) scenarios, the behavior of PCA on reconstructing the original data from the reduced dimensions is similar for all angular separation cases.

2. From the average $RMSE$ of figures 7.2(i), (ii) *and* (iii) for the angular separations 90^0, 60^0 *and* 30^0 at high imbalance ratios, the reprojection error for the minority class is significantly high when compared to the majority class. This evidence indicates that the minority class could not be reconstructed well compared to the majority class. This leads to discriminative information loss from the minority class.

3. At balanced class distributions (50%) case, the reprojection error for both the classes is identical, indicating that both the classes could be reconstructed with equal importance.

4. As the angular separation between the principal axes of two classes decreases, the average $RMSE$ incurred for reconstructing the data from reduced dimension also decreases.

5. From figures 7.3(i), (ii), and (iii) it can be concluded that, except for extreme

unbalance case 1%, the kNN classifier performance over original input space in predicting minority and majority class is considerably well. This may be due to the separability between the two classes of data. In extreme unbalance case (1%), because of lack of data, minority class prediction is considerably low.

6. The overall performance of the kNN classifier for all imbalance ratios over first principal component is considerably less than the performance over the original input space (see figures 7.4(i), (ii) and (iii)). This evidence clearly indicates that the classifier performance degradation is due to loss of discriminative information from the under represented minority class on reduced dimensions. Moreover, the experiments also have revealed that higher imbalance ratio means higher the loss of minority class prediction from under represented class. However, at balanced class distributions this effect is nominal.

7. The 0^0 angular separations is a special case where the principal axis directions of two classes fall into the same principal axis direction. In this case the data is not well separable because minority class overlaps with the majority class. In Fig 7.3 (iv) the kNN classifier performance over original input space reveals that as the degree of class imbalance decreases the performance of the kNN classifier in predicting the minority class increases. Moreover from Fig. 7.2(iv), it can be observed that for all imbalance ratios there is less significant difference between minority class and majority class average $RMSE$. Also the performance of the kNN classifier in predicting the minority class over first principal component is identical to the performance on original data (see figure 7.3(iv) and 7.4(iv)). This may be due to both classes' data falling into the same principal axis.

In summary, whenever there is directional difference between the principal axes of the majority and minority class data, the principal components obtained on unbalanced data for two-class classification problem mostly represent the majority class. This biasing nature towards the majority class maximum variance directions leads to large reprojection errors while reconstructing the minority class as well as performance degradation from minority class prediction. Moreover, it is observed from the balanced cases that balancing the class distributions helped to improve

the minority class reprojection rate as well as classifier prediction towards the minority class.

7.5.2 Experiment-B

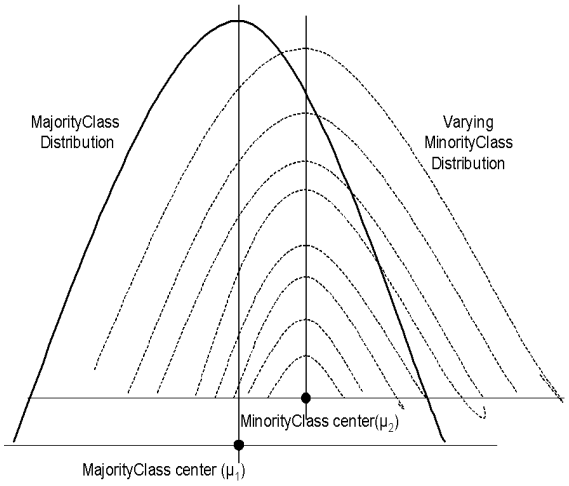


Figure 7.5: Schematic diagram for generating samples with varied degree of overlapping.

In order to analyze the behavior of PCA more deeply, we took the scenario where both classes of data fall along the same principal axis directions (0^0 angular separation case) and Experiment-B was designed. To simulate this behavior, we have synthetically generated 7 datasets with different imbalance ratios (1%, 5%, 10%, 20%, 30%, 40% and 50%) For each dataset, the majority class was centered at (1, 1) with variance of 15, 7 along the axes and the minority class was centered at (12, 3). In addition, for each of these datasets, we have generated 7 datasets with various degrees of overlapping for minority class, by varying the variance from 2, 4, 6, 8, 10, 12 to 14 for I^{st} dimension and 1, 2, 3, 4, 5, 6, 7 for the 2nd dimension. Usually the real world classification problem domains such as fraud detection, medical diagnosis and text classification are constituted with high degree of overlapping of minority and majority class data. The literature on class imbalance problem also proved that the degree of overlapping is one of the key factor for the degradation of classification performance on unbalanced datasets rather than imbalance ratio [5, 91, 47]. Fig 7.5 shows the schematic diagram for generating synthetic samples.

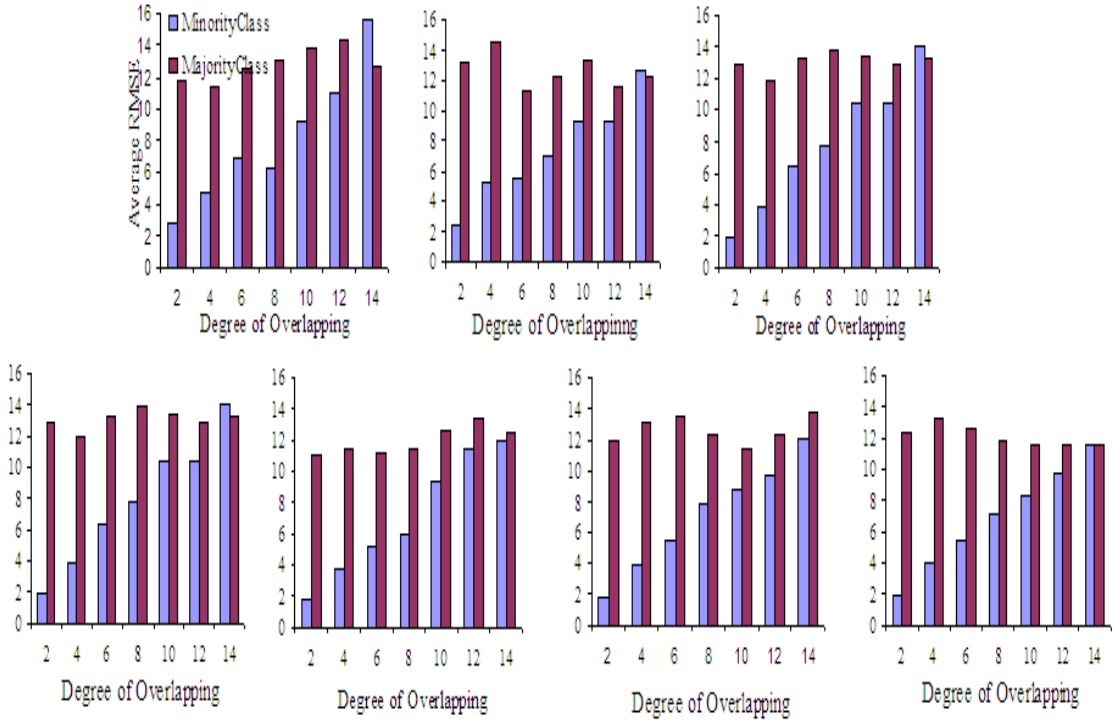


Figure 7.6: Reprojection error with varied degree of overlapping and imbalance ratios. The imbalance ratio is gradually varied from 1% shown at top left panel to 90% shown at bottom right panel

Fig 7.6 and 7.7 show the average reprojection error and classification performance for minority class of these datasets respectively. Fig 7.6 describes the average $RMSE$ for the minority and majority classes over different degrees of overlapping on each degree of imbalance. All these $RMSE$ contributions are obtained on reduced dimensions from 2 dimensional data to 1 dimensional using PCA. The x- axis of Fig 7.6 represents the different degrees of overlapping and the y-axis represents the average $RMSE$. The x-axis of Fig 7.7 represents different degrees of overlapping from the minority class and the y-axis represents the prediction of minority class in terms of $F - measure$. Fig 7.7(i) shows the results on original two-dimensional data, whereas Fig. 7.7 (ii) is for PCA reduced dimension (one-dimension). The observations from these experiments are given below:

1. Fig 7.6 reveals that higher the degree of overlapping on the minority data, larger is the reprojection error. This is true for all the datasets with different imbalance ratios (see Fig 7.6(i) and 7.6(vii)). Further, for each imbalance ratio, at lower degrees of overlapping, minority class average $RMSE$ is significantly less than that of majority class, and at higher degrees of overlapping the average $RMSE$ of the minority class is identical to that of the majority class average $RMSE$. This indicates that in case both classes fall along the same principal axis direction, minority class can be reconstructed well from the reduced dimensions of balanced as well as unbalanced class distributions. Therefore there is no loss of discriminative information from the minority class because of its under-representation.

2. From Fig 7.7(i), it can be observed that as the degree of overlapping increases, the performance of kNN classifier over original unbalanced input space decreases. In addition, the classification performance increases in predicting the minority class as the degree of imbalance decreases.

3. From Fig 7.7(i) and 7.7(ii) it is noticeable that the behavior of kNN classifier in predicting the minority class on PCA reduced dimension (see fig 7.7(ii)) is not consistent compared with the behavior on original input space (see fig 7.7 (i)), for instance, in the case of imbalance ratio 20:80, several variations can be observed: (a) at 2 degree of overlapping case the minority class $F - measure$ on PCA reduced dimension is comparatively less than the

minority class $F-measure$ on original dataset; (b) at 8 degree of overlapping case, the minority class $F-measure$ on PCA reduced dimension is relatively high compared to the minority class $F-measure$ of original dataset; and (c) at 14 degree of overlapping case, the minority class $F-measure$ on PCA reduce dimension is about the same as that of the minority class $F-measure$ of original dataset. The variations in classification performance on PCA reduced dimension are due to the nature of local distribution of the data and how it is represented with reduced dimensions.

4. From Fig 7.6 and 7.7(ii) it is observed that PCA over balanced class distributions is neither giving any additional advantage in reconstructing the minority class data nor in classification performance over the reduced dimensions.

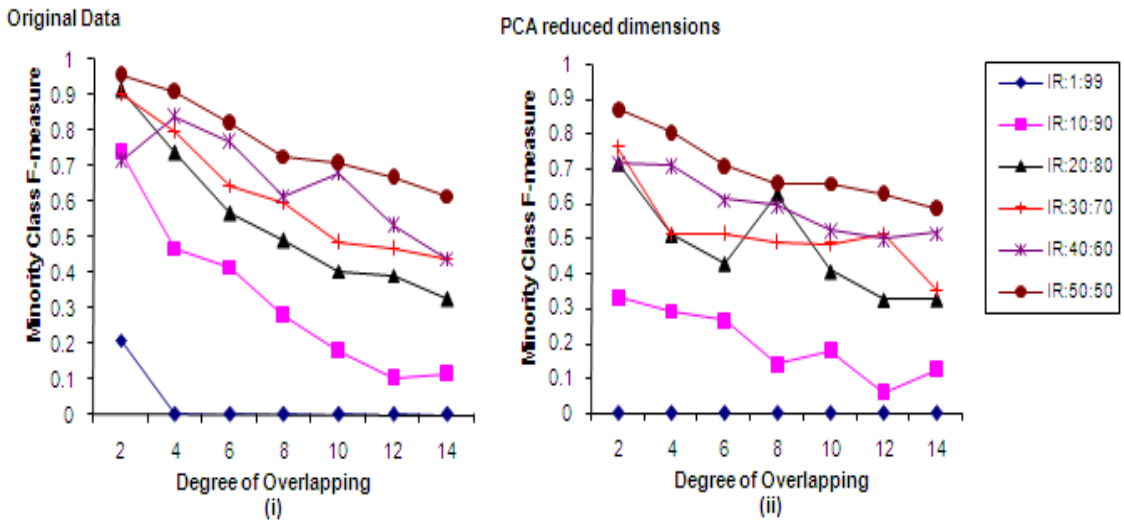


Figure 7.7: Minority class prediction over different degrees of overlapping and imbalance ratios (IR).

Thus, from experiment-B, it is being observed that there is no concrete effect in terms of discriminative information loss from the minority class because of its under-representation on PCA reduced dimensions. Also, from classification point of view, the performance of the classifier learned on PCA reduced dimensions is not prominent in predicting the minority class. Hence learning a classifier on reduced PCA subspace does not give any added advantage or disadvantage for the

class imbalance data where the principal axis direction for both the classes are collinear.

7.6 Experiments on Real World Datasets

The goal of these experiments is to investigate the impact of PCA based dimensionality reduction over unbalanced data that are generated from real-world datasets. These experiments are carried out in line with Experiment-A and Experiment-B described in the previous section. This series of experiments constitute two parts: (i) effect of class imbalance datasets on PCA subspace, and (ii) comparing the classification performance in reduced dimension space for unbalanced and corresponding balanced datasets. Three classifiers with local and global characteristics are considered for analyzing the effect of real-world datasets for two -class unbalance problem.

7.6.1 Datasets

Ten datasets from the UCI repository [9] were used for the experimentation. Out of these datasets, *Pima* dataset is meant for two class classification problem of Diabetes, which describes whether a patient has diabetes or not. The rest of the datasets have more than two classes, and so they are converted into binary class datasets by considering the class with fewer samples as minority class and rest of the samples as majority (negative) class, as suggested in [5]. Table 7.1 shows the description of the datasets used for the experiments. The experiments are carried out over original unbalanced dataset as well as the corresponding balanced datasets using re-sampling techniques.

7.6.2 Experimental Results and Discussion

This section depicts the experimental setup for the series of experiments carried out on real world datasets. As a first step, PCA was directly applied on original unbalanced datasets. Performance on the transformed (reduced dimension) feature space was tested using various classifiers. Next, PCA is applied on balanced datasets and the performance was tested on the same classifiers. Here the balanced class distributions was generated using resampling techniques. Fig 7.8

Table 7.1: Dataset Descriptions

Dataset Number	Datasets	Number examples	Number attributes	Class labels (MAJ-MIN)	Class (MAJ%-MIN%)
1	Flag	194	28	Remainder-White	91.24-8.76%
2	Pima	768	8	1-0	65.1%-34.90%
3	Yeast	1,484	8	Remainder-EXC	97.65%-2.35%
4	Post-operative	87	8	Remainder-S	72.40%-27.60%
5	Waveform	5,000	21	Remainder-1	67.06%-32.94%
6	Image	2,310	18	Remainder-BRICKFACE	89.71%-14.29%
7	Satimage-3	6,435	36	Remainder-3	78.9%-21.1%
8	Iris	150	4	Remainder-Iris-virginica	66.66%-33.33%
9	Wine	178	13	Remainder-3	75.03%-26.97
10	Letter_a	20,000	16	Remainder-a	96.06%-3.94%

describes the flow diagram for the series of experiments that were carried out using resampling techniques, namely, random undersampling, random oversampling and SMOTE.

As shown in Fig. 7.8, the experiments are numbered to identify the type of dataset used for learning a classifier as follows: original unbalanced data (OD) 1, PCA subspace of original data
(OD+PCA) 2, oversampling through generating synthetic samples (SMOTE) 3, PCA subspace of synthetically sampled data (SMOTE+PCA) 4, randomly undersampled data (RUS) 5, PCA subspace of randomly undersampled data (PCA+RUS) 6, randomly oversampled data (ROS) 7 and PCA subspace of randomly oversampled data (ROS+PCA) 8.

The classifiers used for the study are DT, kNN and NB. *recall* (r), *precision* (p) and $F-measure$ (F) are used for evaluating the classifier performance over bal-

anced and unbalanced datasets. The corresponding classification results obtained on series of experiments are reported in Tables 7.2, 7.3 and 7.4 for kNN, DT and NB classifiers, respectively. In this work, for each combination of techniques for balanced data, original unbalanced data and/or reduction in dimension. The best results of classification performance are shown in bold in the tables. A dataset (OD, OD+PCA, RUS, RUS+PCA, ROS, ROS+PCA, SMOTE, SMOTE+PCA) which attained best performance on at least two classifiers is reported as having performed well.

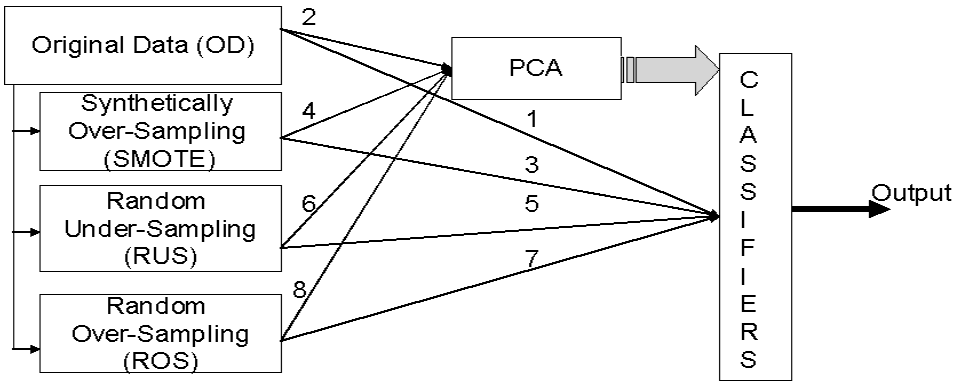


Figure 7.8: Block diagram showing combination of experiments. The numbers on the arrows indicate the experiment number and the same is used in Tables 7.2-7.4

The Effect of Unbalanced Datasets on PCA preprocessing

The results from experiments 1 and 2 (presented in Tables 7.2, 7.3 and 7.4) are being considered here. These experiments depict the results of classifier performance over minority class data on original input space and their corresponding PCA subspace, and our observations are as follows:

- In all experiments, it is clear that the classification results are better only after balancing the distribution, indicating again the importance of balancing for the skewed distributions.

- It is observed that if particular balancing technique (SMOTE, RUS or ROS) gives good results, then the results are equally good whether the PCA applied or not with that balancing technique.

- PCA subspace over $Flag$, $Pima$, $Wine$, $Letter\_a$ and $post\_operative$ datasets have attained equal or better prediction for the minority class with two or

more classifiers when compared to original unbalanced input.

- In the case of $Yeast$, $Waveform$, $Image$, $Satimage-3$ and $Iris$ datasets, the minority class prediction is not improved by 2 when compared to 1. (See Tables 7.2-7.4)

- The minority class $F-measure$ for $Yeast$ and *post_operative* is quite less than 0.5 which indicates highly skewed and overlapping distributions [5, 63]. (See Tables 7.2-7.4). The minority class prediction on 2 also as similar to 1.

For elucidation of PCA analysis over unbalanced classification problem, we computed average angular separation results (Table 7.5) between the top most principal axes of minority and majority classes. The angular separation value near 1 means that two PC projections are in identical directions and their directional separation increases as the value goes to zero. From Table 7.5, it can be observed that the PC directions are similar for most of the datasets. In the case of Image, Iris and Waveform datasets, the PC directions are far apart. (results for these datasets are shown in Tables 7.2-7.4 for OD and OD+PCA cases).

The Effect of Balanced Datasets on PCA preprocessing

This subsection explains the effect of re-sampling techniques for balancing the datasets on PCA subspace. The results from experiments 3 to 8 (presented in Tables 7.2, 7.3 and 7.4) are considered here. These experiments depict the classification results obtained on balanced class distributions and the corresponding PCA subspace. From experiments 3, 5 and 7 on balanced datasets we have observed the following:

- Minority class prediction, in terms of *precision*, *recall* and $F-measure$, obtained on balanced class distribution over three classifiers attained superior performance than on unbalanced data.

- Oversampling methods perform better than the undersampling method for minority class prediction for the considered datasets (See Tables 7.2-7.4). In most of the cases, re-sampling using SMOTE attained better performance than other re-sampling methods (see Tables 7.2 and 7.4).

Table 7.2: *kNN* classification results on original input and reduced feature space over unbalanced and balanced datasets

Data set		OD (1)	OD + PCA (2)	SMOTE (3)	SMOTE + PCA (4)	RUS (5)	RUS + PCA (6)	ROS (7)	ROS + PCA (8)
Flag	P	0	0.059	**1**	**1**	0.412	0.667	1	1
	R	0	0.25	**0.739**	**0.754**	0.7	0.118	0.723	0.72
	F	0	0.095	**0.85**	**0.86**	0.519	0.2	0.84	0.837
Pima	P	0.633	0.646	**0.84**	**0.838**	0.704	0.678	0.768	0.756
	R	0.541	0.552	**0.965**	**0.978**	0.709	0.69	0.931	0.96
	F	0.584	0.596	**0.898**	**0.902**	0.706	0.684	0.842	0.838
Yeast	P	0.543	0.543	**0.997**	**0.998**	0.886	0.886	0.959	0.959
	R	0.594	0.633	**0.932**	**0.94**	0.969	0.939	1	1
	F	0.567	0.585	**0.964**	**0.968**	0.925	0.912	0.979	0.979
Post-operative	P	0	0.125	**0.925**	**0.933**	0.458	0.583	0.458	0.563
	R	0	0.214	**0.81**	**0.762**	0.393	0.519	0.415	0.435
	F	0	0.158	**0.864**	**0.839**	0.423	0.549	0.436	0.491
Waveform	P	*0.832*	*0.732*	**0.996**	**0.985**	0.916	0.876	0.969	0.957
	R	*0.815*	*0.72*	**0.873**	**0.876**	0.88	0.878	0.865	0.827
	F	*0.824*	*0.726*	**0.93**	**0.927**	0.898	0.877	0.914	0.877
Image	P	*0.988*	*0.979*	1	1	**1**	**1**	1	0.874
	R	*0.97*	*0.926*	0.946	0.942	**1**	**1**	0.909	1
	F	*0.979*	*0.951*	0.972	0.97	**1**	**1**	0.952	0.933
Sat-image-3	P	0.714	0.62	1	1	**1**	**1**	1	1
	R	0.704	0.72	0.871	0.831	**0.997**	**0.997**	0.881	0.86
	F	0.709	0.666	0.931	0.912	**0.998**	**0.998**	0.937	0.925
Iris	P	*0.92*	*0.96*	0.97	0.99	**1**	**1**	0.94	0.99
	R	*0.93*	*0.85*	0.97	0.96	**1**	**1**	0.94	0.943
	F	*0.92*	*0.90*	0.97	0.975	**1**	**1**	0.94	0.966
Wine	P	1	1	1	0.99	**1**	**1**	0.96	0.979
	R	0.923	0.979	0.95	0.96	**1**	**1**	1	0.922
	F	0.96	0.959	0.975	0.974	**1**	**1**	0.98	0.949
Letter_a	P	0.991	0.98	**0.997**	**1**	1	0.997	**1**	**1**
	R	0.994	0.999	**0.997**	**0.999**	0.995	0.997	**0.998**	**0.997**
	F	0.992	0.98	**0.997**	**0.999**	0.997	0.997	**0.999**	**0.999**

- As the size of the dataset increases, the effect of unbalancedness on original unbalanced class distribution decreases. For instance, in the case of $post - operative$ dataset which is smaller in size, the classification performance is poor on original unbalanced class distribution, whereas in the case of $Letter - a$ dataset which is larger in size the classification performance is above 0.9 for unbalanced class distribution.

- Taking all classifiers into consideration, decision tree classifier outperforms in terms of exhibiting best $F - measure$ from minority class prediction.

From classification results on balanced datasets over PCA subspace (Experiments 4, 6 and 8), we observed the following:

- Three classifiers attained superior performance for minority class prediction in terms of $precision$, $recall$ and $F - measure$, which are obtained on balanced class distribution in PCA subspace.

- From the table 7.5, it is concluded that balancing the class distributions does not effect the principal axis directions for those datasets. However, in case of RUS for Image dataset, the directional difference has increased, perhaps due to the loss of valuable information from the majority class.

- The superior performance of balanced class PCA subspace is not consistent for all re-sampling techniques. Depending upon the distribution of the data on the reduced dimensions the corresponding balanced data PCA subspace resulted in better performance.

- Decision Tree classifier learned with samples from balanced class PCA subspace attained better performance in terms of $recall$ and $F - measure$ than other classifiers.

- Our intuition was that the classification is easier if the principal directions of the majority and minority classes are well separated. These seems to be evidence for this in Table 7.5 for the following datasets, waveform, Image and iris all of which have low value for cosine angle. The classification results for these datasets are good even without balancing across all classifiers.

In summary, experimental study over real-world balanced and unbalanced datasets reveals that whenever there is a directional difference between the two classes' principal axes, the effect of class imbalance on PCA subspace is prominent in terms of reprojection error as well as classification performance in predicting the minority class. PCA subspace on balanced class distribution can give better results in this case. On the other hand, when there is no directional difference between the two classes' principal axes, there is no significant effect of class imbalance on PCA for minority class reprojection as well as classifier performance. Moreover, PCA subspace on random undersampling is not a better choice for classification because of loss of important majority class information that can lead to change in the directions of the original class distribution. The performance improvement in balanced class PCA subspace may be due to the in-built characteristics of noise handling by PCA.

7.7 Chapter Summary

Principal Component Analysis is a widely used unsupervised dimensionality reduction technique for the classification of high dimensional datasets. In this chapter, an empirical study is carried out to study the behavior of PCA on two-class imbalance datasets. The initial conclusions are obtained on synthetic datasets and further validated over real-world datasets.

- This study proved the fact that when there is a directional difference between the minority class and majority class principal axes, the classification performance and reprojection error are affected in PCA reduced subspace.
- Furthermore, when both minority and majority classes fall along the same principal axes direction (collinear), the effect of unbalanced datasets on PCA subspace is not prominent in terms of minority class reprojection error, cosine angle as well as in terms of minority class prediction.
- Similar results have been reported for balanced and unbalanced real-world datasets. Further, there is no impact on balanced class distributions in PCA subspace as well as on classification performance and random undersampled data leads to different directions from the original unbalanced data.

- Estimating cosine angle between PCs may be a good aid for evaluating the viability of preprocessing using PCA for real-world unbalanced datasets.

Table 7.3: Decision tree classification results on original input and reduced feature space over unbalanced and balanced datasets

Dataset		OD (1)	OD + PCA (2)	SMOTE (3)	SMOTE + PCA (4)	RUS (5)	RUS + PCA (6)	ROS (7)	ROS + PCA (8)
Flag	P	0	0.118	0.909	0.968	0.824	0.412	**1**	**1**
	R	0	0.2	0.924	0.892	0.737	0.585	**0.863**	**0.895**
	f	0	0.148	0.916	0.928	0.778	0.483	**0.926**	**0.944**
Pima	P	0.673	0.578	0.87	0.872	0.716	0.711	**0.824**	**0.815**
	R	0.56	0.671	0.804	0.772	0.705	0.735	**0.96**	**0.975**
	F	0.611	0.621	0.836	0.819	0.711	0.723	**0.887**	**0.889**
Yeast	P	0.486	0.4	0.966	0.974	0.771	0.857	**0.979**	**0.98**
	R	0.68	0.452	0.954	0.952	0.794	0.938	**1**	**1**
	F	0.567	0.424	0.96	0.963	0.783	0.896	**0.989**	**0.99**
Post-operative	P	0	0	**0.917**	**0.933**	0.292	0.373	0.417	0.167
	R	0	0	**0.786**	**0.767**	0.467	0.375	0.455	0.421
	F	0	0	**0.846**	**0.842**	0.359	0.375	0.435	0.239
Waveform	P	*0.854*	*0.806*	0.935	0.971	0.886	0.957	**0.987**	**0.993**
	R	*0.813*	*0.719*	0.912	0.919	0.867	0.885	**0.916**	**0.929**
	F	*0.833*	*0.798*	0.924	0.944	0.876	0.92	**0.951**	**0.96**
Image	P	*0.973*	*0.958*	0.99	0.976	**0.967**	**1**	1	1
	R	*0.979*	*0.946*	0.972	0.953	**1**	**0.968**	0.963	0.968
	F	*0.976*	*0.952*	0.981	0.965	**0.983**	**0.984**	0.981	0.984
Sat-image-3	P	0.868	0.692	0.964	0.969	**0.986**	**0.981**	1	1
	R	0.667	0.447	0.942	0.935	**0.979**	**0.99**	0.881	0.949
	F	0.676	0.543	0.953	0.952	**0.982**	**0.986**	0.937	0.974
Iris	P	*0.92*	*0.88*	0.97	0.98	**1**	**0.98**	0.98	0.98
	R	*0.902*	*0.88*	0.951	0.97	**0.98**	**1**	0.95	0.97
	F	*0.911*	*0.88*	0.96	0.975	**0.99**	**0.99**	0.966	0.975
Wine	P	0.917	0.917	0.979	0.906	**0.958**	**0.958**	0.99	0.948
	R	0.917	0.917	0.979	0.906	**0.956**	**0.958**	0.99	0.948
	F	0.863	0.978	0.969	0.926	**1**	**1**	0.979	0.929
Letter_a	P	0.937	0.913	**1**	**0.997**	0.994	0.971	1	1
	R	0.963	0.939	**0.99**	**0.997**	0.991	0.977	0.997	0.995
	F	0.95	0.925	**0.999**	**0.997**	0.992	0.974	0.998	0.997

Table 7.4: Naïve Bayes classification results on original input and reduced feature space over unbalanced and balanced datasets

Dataset		OD (1)	OD + PCA (2)	SMOTE (3)	SMOTE + PCA (4)	RUS (5)	RUS + PCA (6)	ROS (7)	ROS + PCA (8)
Flag	P	0.353	0.294	**0.85**	**0.947**	0.647	0.714	0.847	0.706
	R	0.111	0.119	**0.736**	**0.855**	0.733	0.588	0.637	0.563
	C	0.169	0.169	**0.789**	**0.898**	0.688	0.645	0.727	0.627
Pima	P	0.699	0.67	**0.824**	**0.842**	0.724	0.753	0.811	0.799
	R	0.604	0.552	**0.845**	**0.832**	0.687	0.672	0.746	0.742
	F	0.635	0.605	**0.835**	**0.837**	0.705	0.71	0.777	0.769
Yeast	P	0.771	0.8	**0.91**	**0.92**	0.861	0.886	0.878	0.882
	R	0.321	0.226	**0.912**	**0.93**	0.886	0.861	0.885	0.912
	F	0.454	0.352	**0.911**	**0.925**	0.838	0.873	0.881	0.897
Post-operative	P	0.125	0.125	**0.783**	**0.867**	0.333	0.375	0.188	0.313
	R	0.6	0.333	**0.746**	**0.693**	0.533	0.375	0.6	0.577
	F	0.207	0.182	**0.764**	**0.77**	0.41	0.375	0.286	0.405
Waveform	P	*0.958*	*0.862*	**0.978**	**0.989**	0.993	0.792	0.977	0.989
	R	*0.704*	*0.765*	**0.876**	**0.859**	0.808	0.934	0.866	0.851
	F	*0.811*	*0.79*	**0.924**	**0.92**	0.891	0.857	0.919	0.915
Image	P	*0.988*	*0.948*	0.848	0.995	**1**	**1**	0.994	0.967
	R	*0.468*	*0.702*	0.986	0.981	**1**	**1**	0.829	0.93
	F	*0.635*	*0.807*	0.912	0.988	**1**	**1**	0.904	0.948
Sat-image-3	P	0.872	0.804	0.891	0.9	**0.963**	**0.946**	0.886	0.88
	R	0.337	0.334	0.951	0.793	**0.992**	**0.967**	0.807	0.723
	F	0.487	0.472	0.871	0.843	**0.977**	**0.956**	0.845	0.823
Iris	P	*0.872*	*0.99*	0.98	0.99	**1**	**1**	0.97	0.98
	R	*0.487*	*0.852*	0.899	0.853	**1**	**1**	0.874	0.845
	F	*0.923*	*0.916*	0.938	0.917	**1**	**1**	0.919	0.907
Wine	P	0.979	0.938	1	0.948	1	**0.979**	**1**	0.958
	R	0.922	0.978	0.97	0.989	**1**	**0.922**	0.941	0.989
	F	0.949	0.957	0.985	0.968	**1**	**0.949**	0.97	0.974
Letter_a	P	0.858	0.942	0.856	0.876	0.879	0.937	**0.875**	**0.892**
	R	0.807	0.843	0.882	0.976	0.913	0.945	**0.974**	**0.903**
	F	0.832	0.893	0.869	0.923	0.896	0.941	**0.992**	**0.897**

Table 7.5: Cosine angle values between minority class and majority class top most eigen vectors.

Datasets	Original	SMOTE	RUS	ROS
Flag	0.999	0.999	0.999	0.999
Pima	0.999	0.999	0.998	0.999
Yeast	0.964	0.963	0.937	0.964
Post-operative	0.933	0.932	0.979	0.933
Waveform	0.781	0.781	0.704	0.781
Image	0.164	0.164	0.982	0.164
Satimage-3	0.923	0.9178	0.988	0.923
Iris	0.792	0.792	0.711	0.792
Wine	0.999	0.999	0.998	0.999
Letter_a	0.932	0.924	0.940	0.932

Chapter 8

A Class-Specific Dimensionality Reduction Framework: CPC_SMOTE

As concluded in previous chapter 7 that in case of PCA preprocessing on unbalanced datasets the directional difference between principle axes can affect the minority class prediction as well minority class information loss in PCA subspace. This chapter provides a class-specific dimensionality reduction framework CPC_SMOTE to alleviate the problem discussed in chapter 7. The rest of the chapter is organized as follows. Section 8.2 discusses work related to dimensionality reduction and unbalanced data classification. Section 8.3 provides proposed CPC_SMOTE framework. Section 8.4 presents the experimental evaluation based on a comparative study which is done by applying PCA on whole unbalanced dataset as well as applying SMOTE on unbalanced datasets. Finally, conclusions are given in section 8.5.

8.1 Introduction

For high-dimensional data, classification process may also include a preprocessing step of dimensionality reduction to increase the class discrimination, better data representation and for attaining good computational efficiency. The misclassification rate drastically increases due to spurious dimensions in the original high-dimensional data space, known as the curse of dimensionality problem [36].

Hence there is a need for dimensionality reduction. Principal Component Analysis (PCA) is a popularly used technique for dimensionality reduction. PCA linearly transforms high dimensional data into lower dimensional space by maximizing the global variance of the data as well as minimizing least square error for that transformation. However, PCA is an unsupervised dimensionality reduction technique. Therefore, it is not adequate to hold the discriminative information for classification problems when the maximum variance direction of one class is different from another class i.e., if covariance matrices are unequal [64, 122]. Finding principal axes directions i.e., principle components (PCs) is one of the key steps for PCA and it depends on spread of the data. In case of the unbalanced datasets, the spread is dominated by majority class as its prior probabilities are much higher than minority class samples. Moreover, in real world domains such as intrusion detection systems, the occurrence of intrusion transactions are rare and generating them is a costly process. Mis-predicting these rare intrusion transactions is risky and could lead to financial loss for organizations. Therefore, capturing and validating labeled samples, particularly non-majority class samples in PC subspace for classification task is a challenging issue.

In this chapter, we propose a class-specific dimensionality reduction based oversampling framework named CPC_SMOTE to address class imbalance issue. This is based on Principal Component Analysis subspace where there is directional difference in Principal Components (PCs) of two classes. The proposed framework based on capturing class-specific features in order to hold major variance directions from individual classes and oversampling is to compensate lack of data in the under-represented class. Proposed approach is evaluated over decision tree classifier using *Accuracy* and $F - measure$ as evaluation metrics. Experimental evidence shows that proposed approach yields superior performance on simulated and real world unbalanced datasets compared to classifier learned on reduced dimensions of whole unbalanced datasets as well as on oversampled datasets.

8.2 Related Work

This section describes the work that is related to unbalanced data classification and dimensionality reduction. Villalba and Cunningham [124] evaluate unsupervised dimensionality reduction techniques over one-class classification methods and con-

cluded that Principal Component Analysis (PCA) damages the performance on most of the datasets. Later, Jiang [64] analyzed the role of PCA over unbalanced training sets and concluded that the PCA subspace is biased by the majority class eigen vectors. Furthermore, the authors proposed Asymmetric Principal component Analysis (APCA), a weighted PCA to address the bias issue in PC subspace. In this chapter we propose a class-specific dimensionality reduction based oversampling framework in the context of two-class classification problems. Proposed approach yields superior performance on those datasets where there is directional difference between two classes' principal components.

8.3 The CPC_SMOTE Framework

The main goal behind this framework is to combat class imbalance problem in PCA subspace, where the principal component analysis predominantly represents majority class maximum variance directions only. In order to accomplish this goal a class-specific principal component analysis based oversampling framework is proposed. Fig 8.1 depicts the flow diagram for proposed framework.

The class-specific PCA is for extracting better informative features from both classes, where there is directional difference between two class PCs. This is done by concatenating class-specific principal components that are extracted from each class by applying PCA. Later SMOTE is applied to this reduced feature space of minority class to alleviate the lack of data problem.

The steps for proposed CPC_SMOTE framework are discussed below. Let X_{n*d} be an unbalanced dataset with n number of records and d number of features.

- Step 1: Extract minority class patterns $(M_i)_{p*d}$ and majority class patterns $(M_j)_{q*d}$ from an unbalanced data X_{n*d}, where $n = p + q, q >> p$ and d=number of features in unbalanced data X_{n*d}.

- Step 2: Apply classical PCA on each class, for each class select $r(< d)$ eigen vectors corresponding to the first r large eigen values. Let $(E_c)_{d*r}$ be the corresponding eigen vectors matrices selected in this step. Where $c = 1$ (minority class direction) and 2 (majority class direction).

- Step 3: Concatenate eigen vectors $(E_c)_{d*r}$ obtained in step 2 in order to facilitate class-specific informative directions.

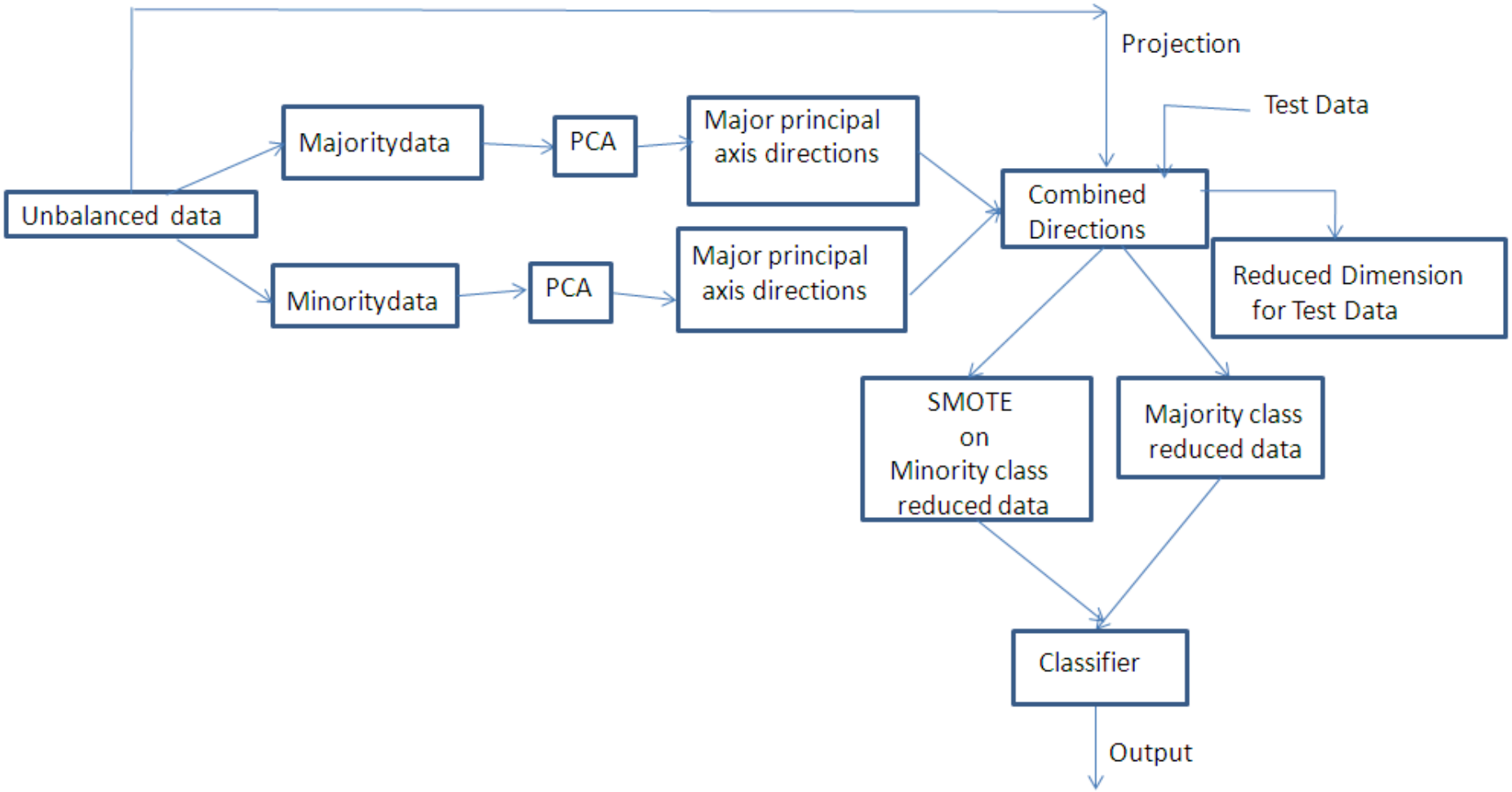


Figure 8.1: Flow diagram for CPC_SMOTE framework.

$(E_{total})_{d*2r} = [(E_1)_{d*r}, (E_2)_{d*r}]$.

- Step 4: Project unbalanced data X_{n*d} into $(E_{total})_{d*2r}$ to enable the informative feature space of majority and minority classes.
  $(NEW\_X)_{n*2r} = X_{n*d} * (E_{total})_{d*2r}$.

- Step 5: Extract reduced minority class $(NEW\_X\_M_i)_{p*2r}$ and majority class $(NEW\_X\_M_j)_{q*2r}$ patterns from the informative feature space $(NEW\_X)_{n*2r}$ and do SMOTE on $(NEW\_X\_M_i)_{p*2r}$.
  $(T\_NEW\_X\_M_i)_{l*2r} = SMOTE(NEW\_X\_M_i)_{p*2r}$,where $l = q$.

- Step 6: Combine minority class and majority class patterns
  $(T\_NEW\_X)_{t*2r} = [(T\_NEW\_X\_M_i)_{q*2r},$
  $(NEW\_X\_Mj)_{q*2r}]$, where $t = 2q$.

In order to get better separated features we apply PCA independently on each class distribution. Let us suppose that $((M_i)_{p*d}, (M_j)_{q*d})$ be the corresponding minority class and majority class distributions, where $q >> p$. From each class distribution of $(M_i)_{p*d}$ and $(M_j)_{q*d}$ equal number of eigen vectors are extracted. Let $r(< d)$ are the selected number of eigen vectors and $(E_c)_{d*r}$ be the corresponding eigen vector matrix, and here $c = 1, 2$. The extracted eigen vectors are combined horizontally to get class-specific features directions $(E_{total})_{d*2r}$. Now the selected number of features becomes $2r$, where r number of reduced features from each class. Then the whole unbalanced data X_{n*d} is projected into the combined feature space $(E_{total})_{d*2r}$ in order to get combined reduced feature space $(NEW\_X)_{n*2r}$. Since the combined class-specific direction matrix $(E_{total})_{d*2r}$ covers the maximum variance directions from all classes, the data can be better discriminated in combined reduced feature space $(NEW\_X)_{n*2r}$. But still there is lack of data problem in reduced feature space of $(NEW\_X)_{n*2r}$ due to the unbalanced nature of the original data. This lack of data problem is alleviated by oversampling the minority class reduced feature space with synthetic samples using SMOTE. In order to do so, minority class reduced feature space $(NEW\_X\_Mi)_{p*2r}$ and majority class reduced feature space $(NEW\_X\_Mj)_{q*2r}$ from $(NEW\_X)_{n*2r}$ are extracted. l number of synthetic samples are generated where $l = (q/p) * p = q$ in minority class feature space $(NEW\_X\_Mi)_{p*2r}$ using SMOTE. Finally both class reduced feature spaces $(T\_NEW\_X\_Mi)_{q*2r}$ and $(NEW\_X\_Mj)_{q*2r}$ are combined to obtain balanced class reduced feature space distribution $(T\_NEW\_X)_{t*2r}$, where

$t = 2q$. The final matrix $(T\_NEW\_X)_{t*2r}$ obtained is the combined matrix with combination of class-specific features and oversampled reduced feature space. This matrix can be directly used for classification tasks. Proposed approach is evaluated using decision tree classifier.

8.4 Experimental Evaluation

This section describes the datasets, presents evaluation metric for estimating classifier performance and elaborates the comparative study with other methods.

8.4.1 Evaluation Metric

Proposed approach is evaluated using classification *Accuracy* (eq. 3.13) and $F - measure$ (eq. 3.17). Classification *Accuracy* is designed to assess the overall classification error rate. Generally, PCA preprocessing on classification process is achieved through improved *Accuracy* on classification process. But in case of unbalanced datasets, *Accuracy* is not an appropriate measure for evaluating the classifier performance. In this chapter *Accuracy* is used for measuring the bias caused by majority class, whereas $F - measure$ is used for evaluating minority class prediction.

8.4.2 Datasets

In order to show how the directional difference between principal axes of PCA affects the performance of unbalanced datasets, we have generated a synthetic dataset. The synthetic dataset are generated using multivariate normal distributions, separately for each class:

Let $p(x)$ be the gaussian probability density function as shown in eq C.2, x is a M-dimensional vector of features, μ is feature vector mean and Σ is a covariance matrix.

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp[-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)] \tag{8.1}$$

The two class distributions are generated with equal mean $\mu = 0$, $M = 10$ uncorrelated features and with unequal covariance matrices. The class covariance matrices are generated in such a way that one class variance dominates the other class variance and with different maximum variance directions, so $(\Sigma_1^2 > \Sigma_2^2)$.

Moreover, generated sample set contains $\omega_1 = 2000$ samples from majority class and $\omega_2 = 100$ samples from minority class with imbalance ratio of 20%. Fig 8.2 depicts the structure of gaussians generated for the evaluation purpose, where ω_1 is the majority class distribution, ω_2 is minority class distribution and probability density of $p(\omega_1) > p(\omega_2)$. The diagonal elements for two classes' covariance matrices for which the off diagonal elements are all zeros are depicted as

$$\Sigma_1 = \begin{pmatrix} 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 \end{pmatrix}$$

$$\Sigma_2 = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 \end{pmatrix}$$

Regarding to real-world datasets, Bronchiolitis dataset is taken from UCD repository [38] and remaining are from UCI repository [9]. Out of these considered data sets, Musk and Bronchiolitis datasets are meant for two-class classification problem. The rest of the datasets have more than two classes, and so they are converted into binary class datasets by considering the class with fewer samples as minority (positive) class and rest of the samples as majority (negative) class, as suggested in [124]. Table 8.1 shows the description of the datasets used for the experiments.

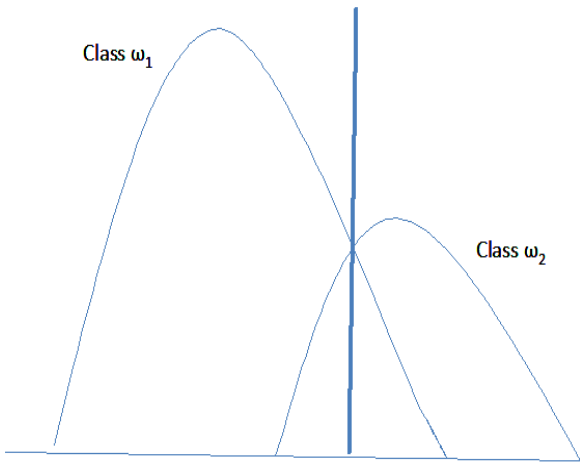


Figure 8.2: Data from classes ω_1 and ω_2 where the probability of $p(\omega_1) > p(\omega_2)$.The vertical line indicate class separation.

Table 8.1: Datasets Description

Dataset	#Min	#Maj	Imb.ratio	#Attri
Simulated	100	2000	20	10
Waveform-1	1647	3353	2	21
Bronchiolitis	37	81	2.18	22
Mfeat-pixel-8	200	1800	9	240
Satimage-4	626	5809	9.27	36
Musk	1017	5581	5.48	166

8.4.3 Experimental Results and Discussion

Since Decision Tree classifier is shown to be sensitive to class imbalance problem, in this work DT is used as a baseline classifier for evaluating the proposed CPC_SMOTE and compared with other data transformations. Decision tree classifier learned on proposed CPC_SMOTE framework is compared with same classifier learned on original data, PCA subspace and on the oversampled data obtained using SMOTE. Experimental evaluation is carried out with the average *F-measure* of 10 fold cross validation on each dataset.

Table 8.2 describes the obtained results on various methods in terms of *Accuracy* and $F-measure$. Here θ identifies the directional difference between the two first principal components of the two classes in terms of cosine angle. If θ is near to one means both classes principal components are in same direction else they are in different directions. The best performance in terms of $F-measure$ that is obtained for that dataset is represented in bold. In table 8.2 $PC(Min_1, Maj_1)$ accounts the variance of first PC's of minority and majority classes. The results on simulated dataset reported 95% *Accuracy* for all methods, but minority class $F-measure$ varied considerably. PCA on simulated dataset performed badly compared with all other methods. Even though the overall *Accuracy* is 95% for the simulated data, the minority class prediction in terms of $F-measure$ is close to 0. This might be because of two reasons. Firstly it may be due to maximum variance coverage from majority class by ignoring minority class maximum variance direction which contributes less variance to the whole distribution. Here θ=0.137 for the simulated data which shows larger directional difference between two class PC's. Secondly, the lack of data causes unnecessary overlap in the PCA subspace. This result clearly substantiates the effect of unbalance data on PCA. Further, classifier learned on original data attained 49% of $F-measure$. As expected SMOTE showed large improvement of 46% over original class $F-measure$. Proposed CPC_SMOTE yielded superior performance among all methods with 95.2% $F-measure$.

Comparing the results of real world datasets, for Waveform dataset CPC_SMOTE yielded superior performance of 92% in terms of both *Accuracy* and $F-measure$ than rest of the methods. The value $\theta = 0.781$ clearly indicates the directional difference between two clases's PC directions. For this dataset PCA showed an improvement of 9.6% $F-measure$ on original data. But compared to the *Accuracy*,

Table 8.2: Comparison of CPC_SMOTE(C_S) performance with Original Data(OD), PCA, SMOTE(S) using Decision Tree classifier

Dataset	θ	$Accuracy\%$				$F-measure\%$				Selected Features		$PC(Min_1, Maj_1)$
		OD	PCA	S	C_S	OD	PCA	S	C_S	PCA	C_S	
Simulated	0.137	95	95	95	95	49.1	*0*	95	**95.2**	4	8	(1.729, 7.349)
Waveform-1	0.781	85	91	88	92	77.4	87	88	**92.2**	5	10	(0.459, 0.693)
Bronchiolitis	0.297	72	65	79.3	78.2	54.3	*42.3*	79.2	**80.2**	7	14	(0.372, 0.394)
Mfeat-pixel-8	0.671	93.6	88.3	96.7	93.5	68.8	*32.8*	**96.9**	94	40	80	(8.895, 7.338)
Satimage-4	0.923	91.8	93.2	94.4	94.6	56.1	61.8	94	**94.4**	7	14	(0.164, 0.754)
Musk	0.901	96.8	97	97.5	96.5	89.8	90.9	**97.5**	96.2	35	70	(7.337, 13.773)

minority class $F-measure$ on PCA is less, showing the bias towards majority class. In this dataset SMOTE did not perform well in improving the performance of original dataset both in terms of $F-measure$ and *Accuracy* than rest of the two methods. For Bronchiolitis and Mfeat-pixel datasets classification *Accuracy* on PCA subspace is not improved compared to *Accuracy* on original data set. Moreover, for these datasets minority class prediction in terms of $F-measure$ is considerably less than the $F-measure$ on original datasets as in the case of simulated dataset. Corresponding directional differences θ=0.297 and 0.671 show larger variation in principal axis directions which substantiates the evidence for performance loss in minority class data due to bias of PCA subspace towards majority class. However, proposed solution CPC_SMOTE yields superior performance on Bronchiolitis dataset with 80.2% minority class prediction in terms of $F-measure$. But on Mfeat-pixel data set SMOTE yielded superior performance than CPC_SMOTE. For this dataset compared with PCA subspace on original data CPC_SMOTE reduces the bias caused by selecting the majority class maximum variance directions. For Satimage dataset even though the two classes PCs represents the same direction with $\theta = 0.923$, classification *Accuracy* and minority class $F-measure$ of CPC_SMOTE is superior to rest of the two methods with 94.7% minority class $F-measure$ and with 94.6% overall *Accuracy*. For this dataset PCA on decision tree classifier showed consistent improvement on the performance on original data in terms of $F-measure$ with 5.7% as well as in total *Accuracy* with 1.4% respectively. For this dataset, CPC_SMOTE showed 38.3% improvement in minority class $F-measure$ over the minority class $F-measure$ of OD.

For Musk dataset where the directional difference $\theta = 0.901$, SMOTE achieved superior performance than rest of the methods with 97.5% minority class prediction and over all *Accuracy* of 97.5%. CPC_SMOTE stood in second position with 96.2% $F-measure$ and 96.5% overall *Accuracy*. PCA showed 1% and 2% consistent improvement on minority class $F-meaure$ and on overall *Accuracy*. Though CPC_SMOTE is computationally costlier than PCA, it is effective in alleviating the bias caused by majority class in PCA subspace and enables better minority class prediction. From the 6 datasets considered CPC_SMOTE outperformed on 4 datasets for which the directional difference is high.

8.5 Chapter Summary

This chapter proposed a class-specific dimensionality reduction and oversampling framework. Proposed framework alleviates the minority class discriminative information loss in PC subspace while selecting maximum variance directions for unbalanced datasets. CPC_SMOTE framework is compared with classical PCA for dimensionality reduction and SMOTE for oversampling in terms of *Accuracy* and $F - measure$. Experimental evidence showed that proposed approach yields superior performance in terms of dimensionality reduction and classification of unbalanced data where the maximum variance predominantly represents majority class.

Chapter 9

Conclusion and Future Work

This chapter concludes the dissertation. A set of conclusions are described in section 9.1. Furthermore, future work is described in section 9.2.

9.1 Conclusions

This thesis has addressed the problem of training with unbalanced datasets for two-class classification in the domain of data mining and machine learning. In the two-class classification problem, one class of data, called the majority class severely outnumbers other class of data called the minority class. The performance of traditional classification techniques such as Decision Tree, Neural networks and Naïve Bayes deteriorates when the techniques are applied on unbalanced datasets. To address the problem this thesis presents five different contributions using generic machine learning concepts such as clustering, undersampling, oversampling, active learning and PCA.

As a first contribution, a hybrid of Extreme outlier elimination + Synthetic Minority Oversampling TEchnique (SMOTE) and Random Under Sampling (RUS) is proposed. The hybrid approach was applied on insurance fraud data sets. To find the extreme outliers, k-Reverse nearest neighbour algorithm was used. In order to demonstrate the hybrid approach, experiments were carried out on classifiers such as Decision Tree (DT), Radial Basis Function (RBF) network Naïve Bayes (NB) and on the k- Nearest Neighbour (kNN). The obtained results were compared with other approaches, namely, SMOTE+ RUS. Experimental evidence shows that ignoring minority class extreme points with the hybrid sampling can

improve the minority class prediction rate (TP rate) and majority class prediction (TN rate). Thus, classifier $G-mean$ is improved. Furthermore, the results indicated that decision tree classifier on proposed hybrid outperformed the rest of the classifiers considered.

The Majority Filter based Minority Prediction (MFMP) approach is the second contribution of this thesis. It constitute two steps. Let S_{min} be the bin for minority class samples. As a first step the minority samples are grouped by clustering process and the majority class samples out of these minority clusters are added to S_{min} and classifier is learned on S_{min}. As a second step, from each minority class cluster whose imbalance ratio is high, majority class samples are randomly selected and added to S until there is an improvement in $F-measure$. Proposed approach is validated on classifiers such as, DT, RBF , NB and kNN classifiers. Experimental evidence indicated that proposed approach yields superior performance than random undersampling. This method is evaluated on minority class $F-measure$.

Undersampling greatly affects the hyperplane orientation of SVM classifier. Therefore, a probabilistic active sampling technique called CStatQSVM is proposed to address the two-class classification problem. The proposed algorithm was designed with a new stopping criteria. The approach was evaluated comparatively over 9 UCI repository benchmark datasets. Wilcoxon Signed-Ranks test as well as Friedman's ranking test reveal that CSTATQSVM is statistically superior to random undersampling and attained Friedman's best mean rank over the rest of the methods.

Principal component analysis is a potential technique for reducing dimensions of unbalanced data. We have evaluated the effect of PCA on unbalanced datasets using the classifiers such as DT, kNN and NB classifiers. PCA was evaluated on two sets of synthetically simulated datasets. The results obtained on simulated data sets were further validated on 10 benchmark datasets. The study concluded that directional difference between the two class principal axes extent of lose of discriminative information from minority class in terms of reconstruction error as well as deterioration in the minority class prediction.

In another contribution, CPC_SMOTE is proposed which is devised based on the conclusions in chapter 7. Chapter 7 concluded that the directional difference in unbalanced data set PC's affects the PCA subspace as well as the two-class

classification performance. The approach was evaluated comparatively with PCA and SMOTE. Results indicated that CPC_SMOTE is better than PCA subspace in improving classifier performance.

Fig 2.3 presents the roadmap of the contributions proposed by this thesis for unbalanced data classification problem and the same is reported in Fig 9.1.

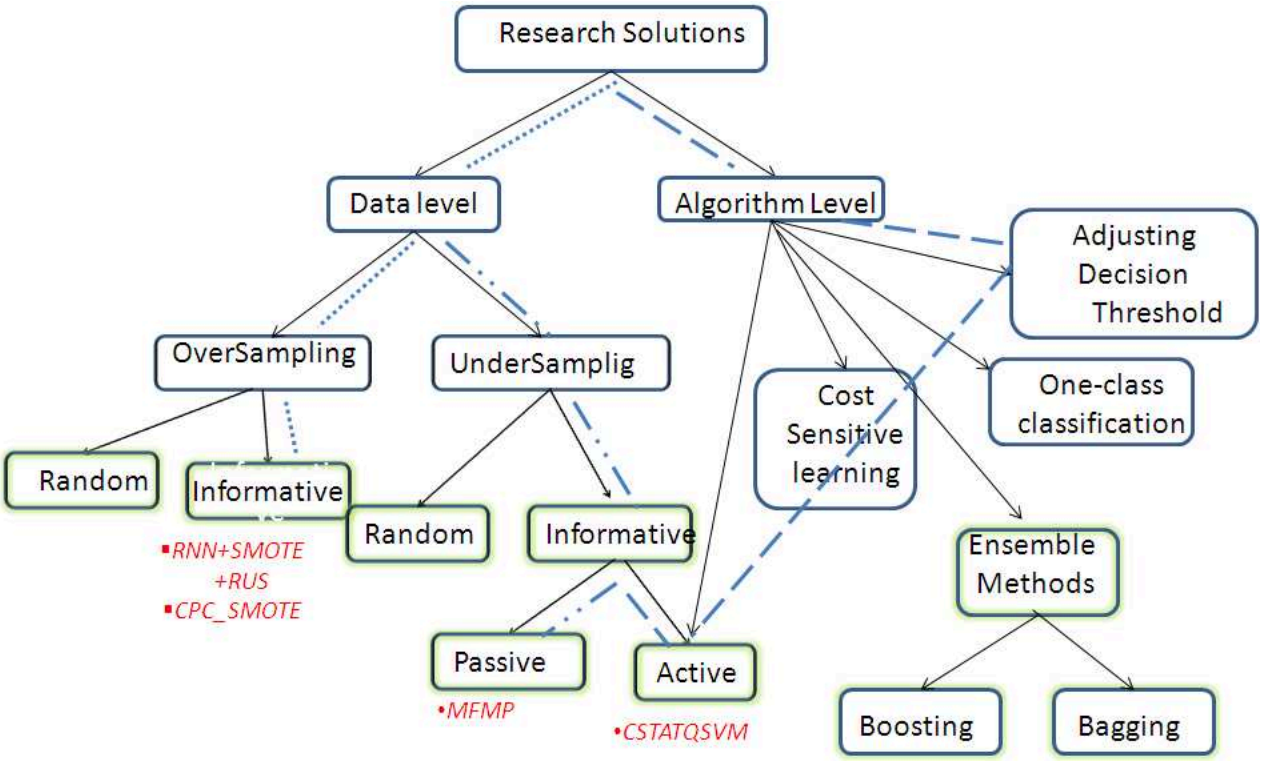


Figure 9.1: Roadmap of the proposed solutions in unbalanced data taxonamy

Therefore, this thesis has achieved its primary goals by providing *new sampling solutions and substantial comparison between existing methods for the unbalanced data classification problem.* In addition to this goal, *this thesis also investigated whether the unbalanced datasets have impact on PCA preprocessing and on the corresponding two-class classification performance.*

9.2 Future Work

The following aspects could be investigated to extend the work reported in the thesis

- The oversampling amd undersamling solutions provided in chapter 4 and 5 are further extendable to multi-class unbalanced data classification problems.
- As the research on multi-class classification abilities of Support Vector Machines is now emerging, the active learning solution provided in chapter 6 should be extended to multi-class SVM active learning for selecting informative instances.
- A preliminary study on whether PCA is effective for preprocessing unbalanced datasets is carried out in chapter 7, but only limited number of studies exist to show the viability of dimensionality reductions techniques to unbalanced datasets. The other dimensionality reduction techniques towards unbalanced datasets on various classifiers need be explored in similar lines.
- As chpater 8 provides a solution to overcome the limitation of PCA preprocessing towards unbalanced datasets, several other solutions based on other variants of dimensionality reduction techniques could also be explored.
- Theoretical justification of the viability of PCA in the lines of described in [52, 141] for LDA should be investigated to get concrete insights.

Bibliography

[1] R. Akbani, S. Kwek and N. Japkowicz, Applying support vector machines to imbalanced datasets. In Proc. of European Conference on Machine Learning, pp: 39-50, 2004.

[2] H. Altinay and C. Ergun, Clustering based under-sampling for improving speaker verification decisions using AdaBoost, In Proc. of Joint IAPR International Workshops on Structural, Syntactic and Statistical Pattern Recognition, (SSPR/SPR04), pp: 698-706, 2004.

[3] R. Anand, K. G. Mehrotra, C. K. Mohan and S. Ranka, An improved algorithm for neural network classification of imbalanced training sets, IEEE Transactions on Neural Networks 4(6), pp: 962-969, 1993.

[4] P. B. Andrew and C. L. Brian, Cost-sensitive decision tree pruning use of the roc curve, In Proc. of the 8^{th} Australian Joint Conference on Artificial Intelligence, Singapore: World Scientific Publ. Co.., pp: 1-8, 1995.

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

[6] R. Brause, T. Langsdorf and M. Hepp, Neural Data Mining for Credit Card Fraud Detection, In Proc. of 11^{th} IEEE International Conference on Tools with Artificial Intelligence, Illinois, USA, pp. 103-106, 1999.

[7] L. Breiman, Bagging predictors, Machine Learning 24(2), pp: 123-140, 1996.

[8] L. Breiman. Random forests. Machine Learning, 45(1), pp: 5-32, October 2001.

[9] C. L. Blake and C. J. Merz, UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science, 1998. $http://archive.ics.uci.edu/ml/$.

[10] M. Bloodgood and K. Vijay-Shanker, Taking into account the differences between actively and passively acquired data: The case of active learning with support vector machines for imbalanced datasets, In Proc. of NAACLHLT'2009, pp: 137-140, 2009.

[11] C. E. Brodley and M. A. Friedl, Identifying mislabeled training data, Journal of Artificial Intelligence Research 11, pp: 131-167, 1999.

[12] C. Campbell, N. Cristianini and A. Smola, Query learning with large margin classifiers, In Proc. of the 17^{th} International Conference on Machine Learning (ICML'00), pp: 111-118, 2000.

[13] P. Chan, W. Fan, A. Prodromidis and S. Stolfo, Distributed Data Mining in Credit Card Fraud Detection, IEEE Intelligent Systems 14(6), pp: 67-74, 1999.

[14] A. L. Chen and L. Breiman, Using random forests to learn unbalanced data, Technical Report 666, Statistics Department, University of California at Berkeley, 2004. $http://stat-www.berkeley.edu/users/chenchao/666.pdf$.

[15] C. Chang and C. Lin, LIBSVM: a library for support vector machines, 2001. Software available at $http://www.csie.ntu.edu.tw/Xcjlin/libsvm$.

[16] J. X. Chen, T. H. Cheng, A. L. F. Chan and H. Y. Wang, An application of classification analysis for skewed class distribution in therapeutic drug monitoring - the case of vancomycin, Workshop on Medical Information Systems (IDEAS-DH'04), pp: 35-39, IEEE. Beijing, China, 2004.

[17] X. W. Chen,B. Gerlach and D. Casasent, Pruning support vectors for imbalanced data classification, In Proc. of the International Joint conference on Neural Networks, pp: 1883-1888, 2005.

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligance Research (JAIR) 16, pp: 321-357, 2002.

[19] N. V. Chawla, C4.5 and Imbalanced Data Sets: Investigating the Effect of Sampling Method, Probabilistic Estimate, and Decision Tree Structure, In Proc of ICML'2003 Workshop on Learning from Imbalanced Data Sets II, August 2003. $http : //www.site.uottawa.ca/ \sim nat/Workshop2003/workshop2003.html$

[20] N. V. Chawla, A. Lazarevic, L. O. Hall and K. W. Bowyer, SMOTEBoost: improving prediction of the minority class in boosting, In Proc.of 7^{th} European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03), Lecture Notes in Computer Science 2838, Springer-Verlag, Cavtat Dubrovnik, Croatia, pp: 107-119, 2003.

[21] N. V. Chawla, N. Japkowicz and A. Kolcz, (eds), In Proc of the ICML'2003 Workshop on Learning from Imbalanced Data Sets-II, August 2003. $http : //www.site.uottawa.ca/ \sim nat/Workshop2003/workshop2003.html$.

[22] N. V. Chawla, N. Japkowicz, and A. Kolcz, (Editorial): Learning from Imbalanced Datasets, SIGKDD Explorations, 6(1), pp: 1-6, 2004.

[23] N. V. Chawla, N. Japkowicz, and Z. Zhou, (eds), In Proc of the PAKDD2009 Workshop on Data Mining when Classes are Imbalanced and Errors Have Costs, August 2009. $http : //www.nd.edu/ \sim dial/Workshop2009/workshop2009.html$.

[24] N. V. Chawla, L. O. Hall and A. Joshi, Wrapper-based Computation and Evaluation of Sampling Methods for Imbalanced Datasets. In Proc.of KDD Workshop, Utility-Based Data Mining, pp: 24-33,2005.

[25] N. V. Chawla, D. A. Cieslak, L. O. Hall and A. Joshi, Automatically Countering Imbalance and Its Empirical Relationship to Cost, Utility-Based Data Mining: A Special issue of the International Journal Data Mining and Knowledge Discovery 17(2), pp: 225-252, 2008.

[26] D. A. Cieslak and N. V. Chawla, Learning Decision Trees for Unbalanced Data, In Proc. of European Conference on Principles and Practice of Knowledge Discovery in Databases, pp: 241-256, 2008.

[27] D. A. Cieslak, N. V. Chawla and A. Striegel, Combating Imbalance in Network Intrusion Datasets, IEEE International Conference on Granular Computing, Athens, Georgia, pp: 732-737, May 2006.

[28] S. Chandana, H. Leung and K. Trpkov, Staging of Prostate Cancer Using Automatic Feature Selection, Sampling and Dempster-Shafer Fusion, Cancer Inform 7, pp: 57-73, 2009.

[29] G. Cohen, M. Hilario H. Sax and S. Hugonnet, Data imbalance in surveillance of nosocomial infections, 4^{th} International Symposium on Medical Data Analysis (ISMDA03), pp: 109-117, 2003.

[30] T. M. Cover, P. Hart, Nearest neighbour pattern classification, IEEE Transactions on Information Theory 13(1), pp: 21-27, 1967.

[31] J. Davis and M. Goadrich, The Relationship between Precision-Recall and ROC Curves, In Proc. of Intenational Conference on Machine Learning, pp: 233-240, 2006.

[32] P. Domingos, MetaCost: A General Method for Making Classifiers Cost-Sensitive, In Proc. of 5^{th} ACM SIGKDD international conference on Knowledge discovery and data mining, pp: 155-164, 1999.

[33] J. Doucette and M. I. Heywoody, GP Classifcation under Imbalanced Data sets: Active Sub-sampling and AUC Approximation, In Proc. of the European Conference on Genetic Programming (Eu- roGP), Lecture Notes in Computer Science, Vol. 4971. Copyright Springer-Verlag, pp: 266-277 , 2008.

[34] C. Drummond, and R. C. Holte, Exploiting the Cost (In) sensitivity of Decision Tree Splitting Criteria, In Proc. of the 7^{th} International Conference on Machine Learning, pp: 239-249, 2000.

[35] C. Drummond and R. C. Holte, C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling, In Proc. of ICML'2003 Workshop on Learning from Imbalanced Data Sets II, August 2003. $http : //www.site.uottawa.ca/ \sim nat/Workshop2003/workshop2003.html$.

[36] R. O. Duda, P. E. Hart and D. G. Stork, Pattern classification and scene analysis, Wiley, New York, 2001.

[37] J. Demsar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7, pp: 1-30, 2006.

[38] D. Doyle, P. Cunningham and P. Walsh, An evaluation of the usefulness of explanation in a case-based reasoning system for decision support in Bronchiolitis treatment. Computational Intelligance 22(3-4), pp: 269-281, 2006. http://mlg.ucd.ie/datasets.

[39] S. Ertekin, J. Huang, L. Bottou and C. L. Giles, Learning on the border: active learning in imbalanced data classification, In Proc. of the 16^{th} ACM Conference on Information and Knowledge Management (CIKM), pp: 127-136, 2007.

[40] A. Estabrooks, T. Jo and N. Japkowicz, A Multiple Resampling Method for Learning from Imbalances Data Sets, Computational Intelligence 20(1), pp: 18-36, 2004.

[41] A. Fernandez, S. Garcia, M. J. del Jesus and F. Herrera, A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets, Fuzzy Sets and Systems 159(18), pp: 2378-2398, 2008.

[42] W. Fan, S. J. Stolfo, J. Zhang and P. K. Chan, AdaCost: misclassification cost-sensitive boosting, In Proc. of 16^{th} International Conference on Machine Learning (ICML'99), Bled, Slovenia, pp: 97-105, 1999.

[43] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27(8), pp: 861-874, 2006.

[44] T. Fawcett and F. Provost, Adaptive fraud detection, Data Mining and Knowledge Discovery 1(3), pp: 1-28, 1997.

[45] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32, pp: 675–701, 1937.

[46] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Annals of Mathematical Statistics 11, pp: 86-92, 1940.

[47] V. Garcia, J. S. Sanchez and R. A. Mollineda, On the $k - NN$ performance in a challenging scenario of imbalance and overlapping, Pattern Analysis and Application 11(3-4), pp: 269-280, 2008.

[48] C. Gathercole and P. Ross, Dynamic training subset selection for supervised learning in genetic programming, In Parallel Problem Solving in Nature III, volume 866 of LNCS, pp: 312-321, 1994.

[49] R. Guha, D. Dutta, P. Jurs and T. Chen, R-NN Curves: An Intuitive Approach to Outlier Detection Using a Distance Based Method, Journal of Chemical Information Modeling 46, pp: 1713-1722, 2006.

[50] H. Guo and H. L. Viktor, Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach, SIGKDD Explorations 6(1), pp: 30-39, 2004.

[51] X. Guo, Y. Yin, C. Dong, G. Yang and G. Zhou, On the class imbalance problem, In proc. of 4^{th} International Conference on Natural Computation (ICNC'08), pp: 192-201, 2008.

[52] J. Hao and M. D. Titterington, Do unbalanced data have a negative effect on LDA?, Pattern Recognition 4(5), pp: 1558-1571, 2008.

[53] J. Han and M. Kamber, Data mining Conceps and Techniques, 2^{nd} ed, Elsevier publisher, 2006.

[54] H. Han, W. Y. Wang and B. H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. International Conference on Intelligent Computing (ICIC'05), Lecture Notes in Computer Science 3644, Springer-Verlag, Hefei China, pp: 878-887, 2005.

[55] P. E. Hart, The Condenced Nearest Neighbour Rule, IEEE Transactions on Information Theory 14(3), pp: 515-516, 1968.

[56] H. He, E. A. Garcia, Learning from Imbalanced Data, IEEE Transactions on Knowledge and Data Engineering 21(9), pp: 1263-1283, 2009.

[57] R. C. Holte and C. Drummond, Cost Curves: An Improved Method for Visualizing Classifier Performance, Machine Learning 65(1), pp: 95-130, 2006.

[58] R. C. Holte and C. Drummond, Cost-Sensitive Classifier Evaluation, In Proc. of International Workshop on Utility-Based Data Mining, pp: 3-9, 2005.

[59] R. C. Holte and C. Drummond, Explicitly Representing Expected Cost: An Alternative to ROC Representation, In Proc. of International Conference on Knowledge Discovery and Data Mining, pp: 198-207, 2000.

[60] R. Holte, N. Japkowicz, C. Ling and S. Matwin, (eds), In Proc. of AAAI2000 Workshop on Learning from Imbalanced Data Sets, AAAI Tech Report WS-00-05, 2000. $http : //www.site.uottawa.ca/ \sim nat/Workshop2000/workshop2000.html$.

[61] H. Ishibuchi and T. Yamamoto, Comparison of heuristic rule weight specification methods, In Proc. of IEEE International Conference on Fuzzy Systems, pp: 908-913, 2002.

[62] N. Japkowicz, C. Myers and M. Gluck, A Novelty Detection Approach to Classification, In Proc. of the 14^{th} Joint Conference on Artificial Intelligence (IJCAI-95), pp: 518-523, 1995.

[63] N. Japkowicz and S. Stephen, The Class Imbalance Problem: A Systematic Study, Intelligent Data Analysis Journal 6(5), pp: 429-449, 2002.

[64] X. D. Jiang, Asymmetric principal component and discriminant analyses for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligance 31(5), pp: 931-937, 2009.

[65] T. Jo and N. Japkowicz, Class Imbalances versus Small Disjuncts, SIGKDD Explorations 6(1), pp:40-49, 2004.

[66] I.T. Jolliffe, Principal Component Analysis, Springer-Verelag, NewYork, 2002.

[67] M. V. Joshi, R. C. Agarwal, V. Kumar, Predicting rare classes: can boosting make any weak learner strong?, In Proc. of 8^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), pp: 297-306, 2002.

[68] F. Korn and S. Muthukrishnan, Influence sets based on reverse nearest neighbour queries, SIGMOD, pp: 201-212, 2000.

[69] M. Kubat, R. Holte and S. Matwin, Learning when Negative Examples Abound, In Proc. of 9^{th} European Conference on Machine Learning (ECML'97), pp:146-153, 1997.

[70] M. Kubat and S. Matwin, Addressing the Curse of Imbalanced Data Sets: One-Sided Sampling, In Proc. of the 14^{th} International Conference on Machine Learning (ICML'97), pp: 179-186, 1997.

[71] M. Z. Kukar and I. Kononenko, Cost-Sensitive Learning with Neural Networks, In Proc. of 13^{th} European Conference on Artificial Intelligence, pp: 445-449, 1998.

[72] P. Kang and S. Cho, EUS SVMs: Ensemble of Under sampled SVMs for Data Imbalance Problems, In Proc. of the 13^{th} International Conference on Neural Information Processing (ICONIP'06), pp: 837-846, 2006.

[73] C. Ling and C. Li, Data Mining for Direct Marketing Problems and Solutions, In Proc. of the 4^{th} International Conference on Knowledge Discovery and DataMining (KDD'98), pp: 73-79, AAAI Press, 1998.

[74] C. Ling, Q. Yang, J. Wang, and S. Zhang, Decision trees with minimal costs, In Proc. of 21^{st} International Conference on Machine Learning, Banff Canada, July 2004. $http://www.machinelearning.org/icml2004\_proc.html$.

[75] H. Lee and S. Cho, Application of LVQ to novelty detection using outlier training data. Pattern Recognition Letters 27(13), pp: 1572-1579, 2006.

[76] H Lee and S Cho, The Novelty Detection Approach for Different Degrees of Class Imbalance, In Proc.of Neural Information Processing, pp: 21-30, 2006.

[77] A. Liu, J. Ghosh and C. Martin, Generative Oversampling for Mining Imbalanced Datasets, In Proc. of International Conference on Data Mining DMIN, pp: 66-72, 2007.

[78] W. Liu, S. Chawla, D. A. Cieslak and N. V. Chawla, A Robust Decision Tree Algorithm for Imbalanced Data Sets, In proc. of SIAM Conference on Data Mining (SDM), pp: 766-777, 2010.

[79] X. Y. Liu, J. Wu and Z. H. Zhou, Exploratory Under Sampling for Class Imbalance Learning, In Proc. of 6^{th} Internatinal conference on Data Mining, pp: 965- 969, 2006.

[80] J. Liu, Q. Hu and D. Yu, A weighted rough set based method developed for class imbalance learning, Information Sciences 178(4), pp: 1235-1256, 2008.

[81] M. Maloof, Learning When Data Sets are Imbalanced and When Costs are Unequal and Unknown, In Proc. of ICML'2003 Workshop on Learning from Imbalanced Data Sets-II, August 2003. $http : //www.site.uottawa.ca/ \sim nat/Workshop2003/workshop2003.html$.

[82] D. Margineantu, Class probability estimation and cost-sensitive classification decisions, In Proc. of 13^{th} European Conference on Machine Learning, Helsinki Finland, pp: 270-281, August 2002.

[83] A. M. Martinez and A. C. Kak, PCA versus LDA, IEEE Transacctiond on Pattern Analysis and Machine Intelligance 23(2), pp: 228-233, 2001.

[84] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker and G. D. Tourassi, Training Neural Network Classifiers for Medical Decision Making: The Effects of Imbalanced Datasets on Classification Performance, Neural Networks 21(2-3), pp: 427-436, 2008.

[85] T. Mitchell, Machine Learning, McGraw Hill, 1997.

[86] P. Mitra, C. A. Murthy and S. K. Pal, A probabilistic active support vector learning algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 26(3), pp:413-418, 2004.

[87] K. Morik, P. Brockhausen and T. Joachims, Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring, In Proc. of the 16^{th} International Conference on Machine Learning, pp: 268-277, 1999.

[88] F. Muhlenbach, S. Lallich and D. A. Zighed, Identifying and Handling Mislabelled Instances, Journal of Intelligent Information Systems 22(1), pp: 89-109, 2004.

[89] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, X. Sun, The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature, Decision Support Systems 50(3), pp:559-569, 2011.

[90] A. Nickerson, N. Japkowicz and E. Milios, Using Unsupervised Learning to Guide Re-Sampling in Imbalanced Data Sets, In Proc. of the 8^{th} International Workshop on AI and Statitsics, pp: 261-265, 2001.

[91] R. C. Prati, G. E. A. P. A. Batista and M. C. Monard, Class Imbalances Versus Class Overlapping: an Analysis of a Learning System Behavior, In Proc. of Mexican International Conference on Artificial Intelligence, LNAI 2972, Springer-Verlag, pp: 312-321, 2004.

[92] R. C. Prati, G. E. A. P. A. Batista and M. C. Monard, Learning with Class Skews and Small Disjuncts, In Proc. of the 17^{th} Brazilian Symposium on Artificial Intelligence, LNAI 3171, Springer-Verlag, pp: 296-306, 2004.

[93] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences 11, pp: 341-356, 1982.

[94] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, Dordrecht, 1991.

[95] C. Phua, D. Alahakoon and V. Lee, Minority report in fraud detection: classification of skewed data, SIGKDD Explorations 6(1), pp: 50-59, 2004.

[96] C. Phua, V. Lee, K. S. Miles and R. Gayler, A Comprehensive Survey of Data Mining-based Fraud Detection Research, Computing Research Repository, abs/1009.6119, 2010. $http://arxiv.org/abs/1009.6119$.

[97] F. Provost and T. Fawcett, Robust Classification for Imprecise Environments, Journal of Machine Learning 42(3), pp: 203- 231, 2001.

[98] J. R. Quinlan, C4.5:Programing for machine learning, Mogan Kaufmann, San Mateo,1995.

[99] B. Raskutti and A. Kowalczyk, Extreme re-balancing for svms: a case study, SIGKDD Explorations Newsletter 6(1), pp: 60-69, 2004.

[100] C. J. V. Rijsbergen, Information Retrieval, Butterworths, London, 2002.

[101] G. Schohn and D. Cohn, Less is more: Active learning with support vector machines, In Proc. of the 17^{th} International Conference on Machine Learning (ICML'00), pp: 839-846, 2000.

[102] R. Schapire and Y. Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning, 37(3), pp: 297-336, 1999.

[103] B. Scholköpf, J. C. Platt, S. J. Taylor, A. J. Smola and R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 113(7), pp: 1443-1471, 2001.

[104] V. Soujanya, V. R. Satyanarayana, K. Kamalakar, A Simple Yet Effective Data Clustering Algorithm, In Proc.of 6^{th} International Conference on Data Mining (ICDM'06), pp: 1108-1112, 2006.

[105] C. Stanfill and D. Waltz, Toward memory-based reasoning, Communications of the ACM 29(12), pp: 1213-1228, 1986.

[106] S. J. Stolfo, D. W. Fan, W. Lee, and A. L Prodromidis, Credit card fraud detection using meta-learning: Issues and initial results, AAAI Workshop on AI Approaches to Fraud Detection and Risk Management, AAAI Press, Menlo Park, CA, pp: 83-90, 1997.

[107] S. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee and D.W. Fan, JAM: Java agents for meta-learning over distributed databases, AAAI Workshop on AI Approaches to Fraud Detection and Risk Management, AAAI Press, Menlo Park, CA, pp: 91-98, 1997.

[108] S. J. Stolfo, W. Fan, W. Lee, A. L. Prodromidis and P. Chan, Cost-based modeling for fraud and intrusion detection: Results from the JAM Project, In Proc of the DARPA Information Survivability Conference and Exposition 2, IEEE Computer Press, New York, pp: 130-144, 1999.

[109] Y. Sun, C. G. Castellano, M. Robinson, R. Adams, A. G. Rust and N. Davey, Using pre and post-processing methods to improve binding site predictions, Pattern Recognition 42(9), pp: 1949-1958, 2009.

[110] Y. Sun, M. S. Kamel, A. K. C. Wong and Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40(12), pp: 3358-3378, 2007.

[111] Y. Sun, A. C. Wong, M. S. Kamel, Classification of Imbalanced Data: A Reviw, International Journal of Pattern Recognition and Artificial Intelligance (IJPRAI) 23(4), pp: 687-719, 2009.

[112] Y. Tang and Y. Q. Zhang, Granular SVM with Repetitive Undersampling for Highly Imbalanced Protein Homology Prediction, In Proc. of the Intenational Conference on Granular Computing, pp: 457-460, 2006.

[113] L. M. Taft, R. S. Evans, C. R. Shyu, M. J. Egger, N. V. Chawla, J. A. Mitchell, S. N. Thornton, B. Bray and M. W. Varner, Countering Imbalanced Datasets to Improve Adverse Drug Event Predictive Models in Labor and Delivery, Journal of Biomedical Informatics (JBI) 42(2), pp: 356-364, 2009.

[114] D. M. J. Tax and R. P. W. Duin, Support Vector Data Description, Journal of Machine Learning 54(1), pp: 45-66, 2004.

[115] J. Thongkam, G. Xu, Y. Zhang and F. Huang, Toward breast cancer survivability prediction models through improving training space, Expert Systems with Applications 36, pp: 12200-12209, 2009.

[116] S. Tong and D. Koller, Support vector machine active learning with applications to text classification, Journal of Machine Learning Research (JMLR) 2, pp: 45-66, 2002.

[117] I. Tomek, Two Modifications of CNN, IEEE Transactions on Systems Man and Communications SMC-6, pp: 769-772, 1976.

[118] K. M. Ting and I. H. Witten, Stacked Generalization: when does it work?, In Proc. of the 15^{th} International Joint conference on Artifical intelligence, pp: 866-871, 1997.

[119] K. M. Ting, A comparative study of cost-sensitive boosting algorithms, In Proc. of 17^{th} International Conference on Machine Learning, Stanford University, CA, pp: 983-990, 2000.

[120] S. Visa, Fuzzy Classifiers for Imbalanced Data Sets, Ph.D Thesis,Univesity of Cincinnati, November 21, 2006.

[121] S. Visa and A. Ralescu, Issues in Mining Imbalanced Data Sets - A Review Paper, In Proc. of the 16^{th} Midwest Artificial Intelligence and Cognitive Science Conference, pp: 67-73, 2005.

[122] N. Vaswani and R. Chellappa, Principal Component Null Space Analysis for Image and Video Classification, IEEE Transactions on Image Processing 15(7), pp: 1816-1830, 2006.

[123] J. R. Vennam and V. Soujanya, Syndeca: A tool to generate synthetic datasets for evaluation of clustering algorithms, In Proc. of the International Conference on COMAD, pp:27-36, 2005.

[124] S. Villalba and P. Cunningham, An evaluation of dimension reduction techniques for one-class classification, Artificial Intelligence Review 27(4), pp:273-294, 2008.

[125] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[126] K. Veropoulos, C. Campbell and N. Cristianini, Controlling the sensitivity of Support Vector Machines, In Proc. of the International Joint Conference on AI, pp: 55-60, 1999.

[127] A. Vlachos, A stopping criterion for active learning, Computer Speech and Language 22(3), pp: 295-312, 2008.

[128] J. Wang, M. Xu, H. Wang and J. Zhang, Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding. In proc. of 8^{th} IEEE International Conference on Signal Processing, pp:16-20, 2006.

[129] B. X. Wang, N. Japkowicz, Boosting Support Vector Machines for Imbalanced Data Sets Knowledge and Information Systems 25(1), pp: 1-20, 2010.

[130] G. M. Weiss, K. McCarthy and B. Zabar, Cost Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs?, In Proc. of the International Conference on Data Mining DMIN, pp: 35-41, 2007.

[131] G. M. Weiss, F. Provost: Learning when training data are costly: the effect of class distribution on tree induction, Journal of Artificial Intelligance Research (JAIR) 19, pp: 315-354, 2003.

[132] G. M. Weiss, The Impact of Small Disjuncts on Classifier Learning, Annals of Information Systems 8, pp: 193-226, 2010.

[133] G. M. Weiss, Mining with Rarity: A Unifying Framework, ACM SIGKDD Explorations 6(1), pp: 7-19, 2004.

[134] R. Wheeler and S. Aitken, Multiple Algorithms for Fraud Detection, Knowledge-Based Systems 3(2/3), pp: 93-99, 2000.

[135] D. L. Wilson, Asymptotic Properties of Nearest Neighbor Rules Using Edited Data, IEEE Transactions on Systems, Man and Communications 2(3), pp: 408-421, 1972.

[136] R. Wilson and T. R. Martinez, Improved Heterogeneous Distance Functions, Journal of Artificial Intelligence Research (JAIR) 6, pp: 1-34, 1997.

[137] I. Witten and E. Frank, Data Mining: Practical Machine Learning tools and Techniques with Java Implementations, Morgan Kaufmann Publishers, 2000.

[138] D. Wolpert, Stacked Generalization, Neural Networks 5(2), pp: 241-259, 1992.

[139] G. Wu and E. Chang, Class-Boundary Alignment for Imbalanced Dataset Learning, In Proc of ICML'2003 Workshop on Learning from Imbalanced Data Sets II, August 2003. $http : //www.site.uottawa.ca/ \sim nat/Workshop2003/workshop2003.html$.

[140] M. Wu and J. Ye, A Small Sphere and Large Margin Approach for Novelty Detection Using Training Data with Outliers, IEEE Transactions on Pattern Analalysis and Machine Intelligance 31(11), pp: 2088-2092, 2009.

[141] J. Xie and Z. Qie, The effect of imbalanced datasets on LDA: A Theoretical and empirical analysis, Pattern Recognition 40(2), pp: 557-562, 2007.

[142] L. Xu, M. Chow and L. Taylor, Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification e-algorithm, IEEE Transactions on Power Systems 22(1), pp: 164-171, 2007.

[143] C. Y. Yang, Jr. S. Yang and J. J. Wang, Margin calibration in SVM class-imbalanced learning, Neurocomputing 73(1-3), pp: 397-411, 2009.

[144] S. J. Yen and Y. S. Lee, Under-Sampling Approaches for Improving Prediction of the Minority Class in an Imbalanced Dataset, In Proc.of the Intelligent Control and Automation (ICIC'06), pp: 731-740, 2006.

[145] K. Yoon and S. Kwek, A data reduction approach for resolving the imbalanced data issue in functional genomics, Neural Computing and Applications 16(3), pp: 295-306, 2007.

[146] J. Yuan J. Li and B. Zhang, Learning Concepts from Large Scale Imbalanced Data Sets Using Support Cluster Machines, In Proc. of the International Conference on Multimedia, pp: 441-450, 2006.

[147] B. Zadrozny, J. Langford and N. Abe, Cost-sensitive learning by cost-proportionate example weighting, In Proc. of 3^{rd} IEEE International Conference on Data Mining, Melbourne, Florida, pp: 435-442, 2003.

[148] Z. Q. Zeng and J. Gao, Improving SVM Classification with Imbalance Data Set, In proc. of Neural Information Processing, pp: 389-398, 2009.

[149] J. Zar, Biostatistical Analysis, Prentice Hall, Upper Saddle River, New Jersey, 1999.

[150] J. Zhang, and I. Mani, KNN approach to unbalanced data distributions: A case study involving information extraction, In Proc. of the ICML'2003 Workshop on Learning with Imbalanced Data Sets II, August 2003. $http://www.site.uottawa.ca/\sim nat/Workshop2003/workshop2003.html$.

[151] Z. H. Zhou and X. Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, IEEE Transactions on Knowledge and Data Engineering 18(1), pp: 63-77, 2006.

Papers Contributed

1. T.Maruthi Padmaja, Narendra Dhulipalla, P.Radha Krishna, Raju S.Bapi, and A.Laha, An Unbalanced Data Classification Model Using Hybrid Sampling Technique For Fraud Detection, In proc. of Pattern Recognition and Machine Intelligence (PreMi), Springer-Verlag, pp: 341-348, ISI-kolkata 2007.

2. T.Maruthi Padmaja, Narendra Dhulipalla and Raju S.Bapi, P.Radha Krishna, Unbalanced Data Classification using extreme outlier Elimination and Sampling Techniques for Fraud Detection, In Proc. of Advanced Computing and Communications (ADCOM), pp: 511-516, IEEE Computer Society Press, IIT-Guwahati 2007.

3. T.Maruthi Padmaja, P.Radha Krishna and Raju S.Bapi, Majority Filter Based Minority Prediction (MFMP) an Approach for Unbalanced Dataset, In Proc. of Annual International Conference Tencon, No:4766705, Hyderabad 2008.

4. T.Maruthi Padmaja, P.Radha Krishna and Raju S.Bapi, Reverse-NN Curve based Cluster Counting Approach, International Conference on Data Management (ICDM), Ghaziyabad 2009.

5. T.Maruthi Padmaja, Raju S.Bapi and P.Radha Krishna, A Class Specific Dimensionality Reduction Framework for Class Imbalance Problem: CPC_SMOTE, In Proc. of Knowledge Discovery and Information Retrieval (KDIR), INSTICC Press, pp: 237-242, Spain 2010.

Book chapter

1. T.Maruthi Padmaja, Raju S.Bapi and P. Radha Krishna, Unbalanced Sequential data classification using Extreme Outlier Elimination and sampling Techniques, Pattern Discovery Using Sequence Data mining: Applications and Studies, IGI Global publisher.

Communicated

1. T.Maruthi Padmaja, Rudra N. Hota, Raju S.Bapi and P. Radha Krishna, Is PCA Effective for Preprocessing Unbalanced Data?, Applied Soft Computing, elsevier, Under review.

2. T.Maruthi Padmaja, Raju S.Bapi and P. Radha Krishna, Probabilistic Cost weighted Active Learning Approach for Class Imbalance Problem, International Journal of Knowledge Engineering and Soft Data Paradigms (IJKESDP), Inderscience, Under review.

3. T.Maruthi Padmaja, Raju S.Bapi and P. Radha Krishna, Unbalanced Data Classification through Outlier Detection: An Application to Fraud Detection, International journal of Data Warehousing and Mining (IJDWM), IGI Global, Under review.

Appendix A

Description of Insurance Fraud Dataset

This Appendix, describe the automobile insurance fraud dataset we have used for experimentation.

A.1 Data Description

The automobile insurance dataset contains 11550 examples from January 1994 to December 1995, and 3870 instances from January 1996 to December 1996. It has a 6% fraudulent and 94% legitimate distribution, with an average of 430 claims per month. The original dataset A.1 has 6 numerical attributes and 25 categorical attributes, including the two category class label (fraud/legal). From a critical viewpoint, the data analyst can lament the lack of very powerful fraud predictors in the form of behavioral attributes such as a claimant's occupation, salary, and education level.

Explore the data

Through extensive querying and visualization of each data attribute, some interesting facts stand out. The number of fraudulent claims underwent a few rise-and-fall cycles. January 1994 to August 1994 accounts for 91% of fraudulent claims for the year 1994. After three quiet months, there was another increase of fraudulent claims from December 1994 to March 1995. April 1995 to August 1995 accounts

for only 4% of fraudulent claims for the year 1995, after which fraud was rife up to December 1995. The consecutive months with dominant illegitimate activities could be due to a wave of hard fraud committed by professional offenders. The data analyst can then hypothesize that fraudulent claims will probably drop for the first few months in the year 1996.The proportion of fraud committed across the age groups for the years 1994 and 1995 is as follows. In both years, middle-aged claimants were responsible for almost 80% of the fraud. But the interesting group is the younger fraudsters between 16 to 25 years old. Although they account for only 6.34% of the total fraud, the proportion of fraud within their age groups is 13%, which is twice the proportion of the whole dataset. Because these younger age groups account for only 3% of all the claims, they are relatively easy to monitor.

A.2 Data Quality

The data quality is good but there are some impediments. The original dataset consists of the attribute *PolicyType* that is an amalgamation of existing attributes *VehicleCategory* and *BasePolicy*. There are invalid values of 0 in each of the attributes *MonthClaimed* and *DayofWeekClaimed* for one example. Some attributes with two categories, like *WitnessPresent*, *AgentType*, and *PoliceReportFiled*, have highly skewed values where the minority examples account for less than 3% of the total examples. The attribute $Make$ has a total of 19 possible attribute values of which claims from Pontiac, Toyota, Honda, Mazda, and Chevrolet account for almost 90% of the total examples. There are three spelling mistakes in Make: *Accura* (Acura), *Mecedes* (Mercedes), *Nisson* (Nissan), and *Porche* (Porsche). Attributes $address\_change - claim$ and $number\_of\_cars$ can have fewer discrete values. For claims made by 16 to 20 year olds, more than 95% of their vehicles are barely one year old and 95% of their insured vehicle prices are above $69,000. This piece of information seems to conflict with common sense because most young drivers cannot afford new and expensive cars.

Table A.1: Original attributes in automobile insurance fraud dataset.

Attribute Name	Description	Categories
Month	Month in which accident took place	12
Week_of_month	Accident week of month	5
Day_of_week	Accident day of week	7
Make	Manufacturer of car	19
Accident_area	City or country	2
Month_claimed	Claim month	2
Week_of_month_claimed	Claim week of month	12
Day_of_week_claimed	Claim day of week	5
Year	1994,1995 and 1996	3
PolicyType		9
Sex	Gender	2
Marital status		4
Fault	Policy holder or other party	2
Vehicle category	Sedan, sport or utility	3
Vehicle price	Price of vehicle	6
Rep_number	ID of person who processed the claim	16
Deductible	Amount to be deducted before claim disbursement	4
Driver rating		4
Days policy-accident	Days left in policy when accident happened	5
Days policy-claim	Days left in policy when claim was filed	4
Past_no_of_claims		4
Age_of_vehilcle		8
Age of policy holder		9
Police_report_filed		2
Witness present		2
Agent type	Internal or external	2
Number of supplements		4
Address change-claim		4
Number of cars		4
Base_policy	All-perils, collision or liability	3
Class	Fraud Found	2

A.3 Data Preparation

This section encompasses the tasks to be done to build the input dataset for the learning algorithms. This phase involves the selecting, cleaning, formatting and construction of the data.

A.3.1 Select the data

Of the original data attributes given (see Table A.1), all have been converted into nominal in advance. For example, instead of a real-valued attribute giving the precise value of the insured vehicle, this dataset includes only a discrete-valued attribute that categorizes this amount into one of six different discrete levels. Most of the data attributes are retained for the data analysis. The only attribute discarded up to this stage is $PolicyType$, which is redundant.

Clean the data

To improve the data quality, the invalid values can be replaced with the majority attribute value. In this case, $MonthClaimed =$ "$January$" and $Day\,of\,Week\,Claimed =$ "$Monday$" replaces the 0 values but the next simple alternative is to delete this example. The four spelling mistakes in the attribute Make are corrected for all the examples. Nothing can be done with the three highly skewed attributes. The number of discrete values in attribute $Make$ is not reduced because certain car brands with few claims can be responsible for high fraud occurrences.

A.3.2 Construct the data

Three derived attributes, $weeks\_past$, $is\_holidayweek\_claim$, and age_price_wsum are created to increase predictive accuracy for the algorithms. The new attribute, $weeks_past$, represents the time difference between the accident occurrence and its claim application. The hypothesis states that if this difference is larger than average, fraud is more likely. The position of the week in the year that the claim was made is calculated from attributes $month\_claimed$, $week\_of\_month\_claimed$, and $year$. Then the position of the week in the $year$ when the accident is reported to have happened is computed from attributes month, $week\_of\_month$, and $year$. The latter is subtracted from the former to obtain the derived attribute $weeks\_past$. This derived attribute is then categorized into eight discrete values.

The derived attribute *is_holidayweek_claim* indicates whether the claim was made in a festive week. The data analyst speculates that average offenders are more likely to strike during those weeks because of the need to increase their spending. The attribute *age_price_wsum* is the weighted sum of two related attributes, *age_of_vehicle* and *vehicle_price*. The premise is that the more expensive and the older the vehicle gets, the possibility of the claim being fraudulent becomes higher. So the attributes used in our experiments are given in the Table A.2.

Table A.2: Modified attributes in the dataset.

Attribute Name	Description	Categories
Month	Month in which accident took place	12
Week_of_month	Accident week of month	5
Day_of_week	Accident day of week	7
Make	Manufacturer of car	19
Accident_area	City or country	2
Month_claimed	Claim month	2
Week_of_month_claimed	Claim week of month	12
Day_of_week_claimed	Claim day of week	5
Year	1994,1995 and 1996	3
Weeks_past	Accident and claim difference	8
Is_holidayweek_claim	Claim was made on holiday week or not	2
Sex Gender		2
Marital status		4
Fault	Policy holder or other party	2
Vehicle category	Sedan, sport or utility	3
Vehicle price	Price of vehicle	6
Rep_number	ID of person who processed the claim	16
Deductible	Amount to be deducted before claim disbursement	4
Driver rating		4
Days policy-accident	Days left in policy when accident happened	5
Days policy-claim	Days left in policy when claim was filed	4
Past_no_of_claims		4
Age_of_vehilcle		8
Age_price_wsum	Age and vehicle price combined	7
Age of policy holder		9
Police_report_filed		2
Witness present		2
Agent type	Internal or external	2
Number of supplements		4
Address change-claim		4
Number of cars		4
Base_policy	All-perils, collision or liability	3
Class	Fraud Found	2

Appendix B

Description of Real-World Datasets

The purpose of this appendix is to describe the datasets used throughout this dissertation. Throughout the body of this work, datasets have been described uniformly. Not every dataset was used for each set of experiments, since some proved inappropriate for certain situations. In other instances, some datasets were not available at the time that set of experiments was performed. The characteristics of each dataset are listed in Table B.1. Nine datasets from the UCI repository [9] were used for the experimentation. Out of these datasets, Ionosphere, Pima and Magic Gamma Telescope (Gamma) datasets were meant for two-class classification problem.

- *Ionosphere* dataset describes the classification of radar returns from ionosphere into good or bad returns. The 'good' radar returns shows the evidence for a type of structure in ionosphere whereas 'bad' return do not show the evidence.
- The *Pima* dataset is about predicting whether a patient has diabetes (1) or not (0).
- The *Magic Gamma Telescope (Gamma)* data set describes the discrimination of primary gamma signals (g) from those images generated by cosmic rays (h) in the upper atmosphere.
- *Haberman*The Haberman dataset is two-class data set about breast cancer patients whether they survive or die within 5 years period of surgery.

- The *Bronchiolitis* dataset is about for predicting whether a patient is infected by bronchiolitis from 0-1 day onwards.

The rest of the datasets (*Glass*, *Waveform*, *Letter-a*, *Satimage*, *Abalone* and *Shuttle*) have more than two classes. The datasets are converted into binary class datasets by considering the class with fewer samples as minority class and rest of the samples as majority class. This approach was suggested by Wu and Edward [139].

- The *Abalone* dataset is of predicting the age between 1 to 29 from physical measurements by cutting the shell through the cone. For this dataset samples with class label '15' considered as minority class and rest of the samples considered as majority class.

- *E_Coli* dataset is of for multi class classification problem. The class of each record in this dataset is a Protein Localization Sites. Among 8 classes in this dataset imU (inner membrane, uncleavable signal sequence) is considered as minority class.

- *Flags* dataset describes the details of various nations and their flags. Here the predictive problem is identifying the religion of a country from its size and the colors in its flag. The white color bottom-left corner is considered as minority class among 7 colors.

- *Yeast* dataset is for predicting the Localization site of Yeast protein. Among 10 Localization sites EXC is chosen as minority class.

- *Postoperative*, Science, hypothermia is a significant concern after surgery this database represents the classification task of, to determine where patients in a postoperative recovery area should be sent to next. The attributes correspond roughly to body temperature measurements. Among 3 attributes, S (patient prepared to go home), selected as minority class.

- *Waveform*, in this dataset class label is selected as minority class among all other labels.

- *Image* The instances for Image segmentation were drawn randomly from a database of 7 outdoor images. Among 7 classes, the class with brickface selected as minority class.

- *Satimage* This database represents Landsat Multi-Spectral Scanner image data. Among 7 classes grey soil(3) picked as minority class.

- *Iris* data set contains 3 classes of 50 instances each, where each class represents a type of iris plant. Here plant type Iris Virginica selected as minority class.

- The *Wine* dataset represents the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis proved that 13 constituents found in each of the three types of wines. In this dataset class label 3 is considered as minority class.

- *Glass* dataset is for the classification of glass types for criminological investigation. In this dataset $vehicle\_windows\_float\_processed$ is picked as minority class.

- *Shuttle* in this dataset High(4)is picked as minority class.

- *Letter_A* dataset is for classifying large number of *black_and_white* rectangular pixel displays as one of the 26 capital letters in the English alphabet. Among 26 alphabets Letter A considered as minority class.

- *Mfeat-pixel* dataset represents the features of handwritten numerals ('0'–'9'). For this dataset the digit with label 8 is chosen as minority class.

- *Musk* This dataset is for predicting whether new molecules will be musks or non-musks. This dataset describes a set of 102 molecules of which 39 are judged by human experts to be musks and the remaining 63 molecules are judged to be non-musks.

Table B.1: Characteristics of Unbalanced Datasets Investigated

Dataset Number	Datasets	Number examples	Number attributes	Class labels (MAJ-MIN)	Class (MAJ%-MIN%)
1	E_Coli [9]	336	7	Remainder-iMU	89%-11%
2	Haberman [9]	306	3	Survive-Die	74%-26%
3	Flag [9]	194	28	Remainder-White	91.24%-8.76%
4	Pima [9]	768	8	1-0	65.1%-34.90%
5	Yeast [9]	1,484	8	Remainder-EXC	97.65%-2.35%
6	Post-operative [9]	87	8	Remainder-S	72.40%-27.60%
7	waveform [9]	5,000	21	Remainder-1	67.06%-32.94%
8	Image [9]	2,310	18	Remainder-BRICKFACE	89.71%-14.29%
9	Satimage-3 [9]	6,435	36	Remainder-3	78.9%-21.1%
10	Iris [9]	150	4	Remainder-Iris-virginica	66.66%-33.33%
11	Wine [9]	178	13	Remainder-3	75.03%-26.97
12	Glass [9]	214	8	(Remainder, Ve-win-float-proc)	92.06%-7.94%
13	Abalone [9]	4177	8	(remainder, 15)	97.56%-2.44%
14	Ionosphere [9]	350	34	(g, b)	64%-36%
15	Gamma [9]	19020	10	(g, h)	64.84%-35.16%
16	Shuttle [9]	58000	9	(remainder, 4)	84.65%-15.35%
17	Letter_a [9]	20,000	16	(Remainder-a)	96.06%-3.94%
18	Bronchiolitis [38]	118	22	(0-1)	68.65%-31.35%
19	Mfeat-pixel-8 [9]	2000	240	(Remainder-8)	90%-10%
20	Musk [9]	6598	166	(pending)	84.51%-15.41%

Appendix C

Description of Synthetic Data Sets used

This Appendix describes the nature of synthetic data sets were generated in each part of this study.

C.0.3 Synthetic Dataset-1

The first synthetic data set used in the study is drawn from uniform distribution ranging from 0 to 100 with two dimensions. Among the specified distribution, 400 samples are drawn for majority class and 50 samples are drawn for minority class. Generated minority class samples [123] are sparsely distributed with respect to the majority class. This dataset is used in chapter 5.

C.0.4 Synthetic Dataset-2

For the second data set, artificial samples are generated with angular separation (θ) between the majority class and minority class principal axes, where $\theta = 90^0,\ 60^0,\ 30^0\ and\ 0^0$. For each of these cases, 7 datasets with different imbalance ratios staring from $1\%, 5\%, 10\%, 20\%, 30\%, 40\% \rightarrow\ 50\%$ were generated. Both majority and minority class are drawn from gaussian distribution,

$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{C.1}$$

centered at (0, 0) with variances 15 and 2 along the axes. Fig C.1depicts the schematic diagram for generation of synthetic samples in different principals axes

direction for majority and minority classes in a two dimensional dataset. In eq C.1 The first parameter, μ, is the mean, the second, σ, is the standard deviation. This dataset is used in chapter 7.

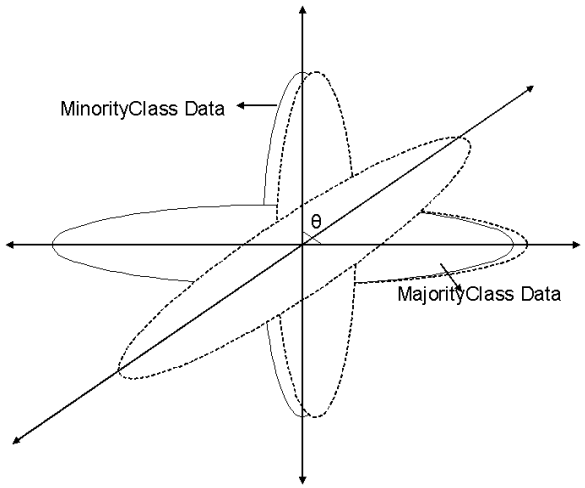


Figure C.1: Schematic diagram for the sample generation in angular separation.

C.0.5 Synthetic Dataset-3

As a third dataset, we have synthetically generated 7 datasets drawn from gaussian distribution, with different imbalance ratios (1%, 5%, 10%, 20%, 30%, 40% and 50%). For each dataset, the majority class centered at (1, 1) with variance of 15, 7 along the axes and the minority class centered at (12, 3). In addition, for each of these datasets, we have further generated 7 datasets with various degrees of overlapping for minority class, by varying the variance from 2, 4, 6, 8, 10, 12 to 14 for I^{st} dimension and 1, 2, 3, 4, 5, 6, 7 for the 2^{nd} dimension. Fig C.2 shows the schematic diagram for generating synthetic samples. This dataset is used in chapter 7.

C.0.6 Synthetic Dataset-4

The fourth synthetic dataset generated using multivariate normal distributions separately for each class:

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp[-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)] \tag{C.2}$$

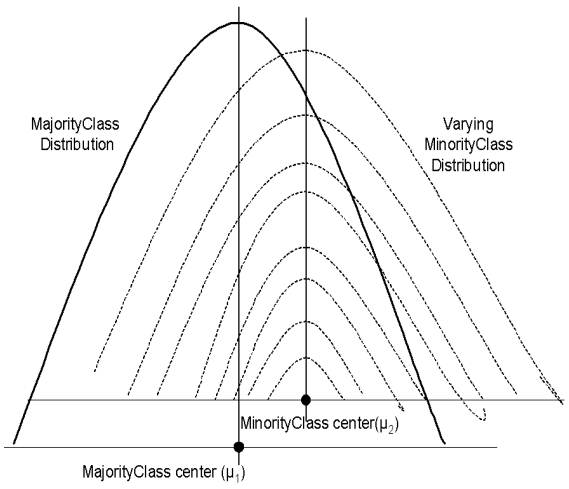


Figure C.2: Schematic diagram for generating samples with varied degree of overlapping.

Where $p(x)$ is probability density function, x is a M-dimensional vector of features, μ is feature vector mean and Σ is a covariance matrix. The two class distribution are generated with equal mean $\mu = 0$, $M = 10$ uncorrelated features and with unequal covariance matrices. The two classes covariance matrices are generated in such a way that one class variance dominates the other class variance and with different maximum variance directions, so $(\Sigma_1^2 > \Sigma_2^2)$. Moreover, generated distribution contains $\omega_1 = 2000$ samples from majority class and $\omega_2 = 100$ samples from minority class with imbalance ratio of 20%. Fig C.3 depicts the structure of gaussians generated for the evaluation purpose, where ω_1 is the majority class distribution, ω_2 is minority class distribution and probability density of $p(\omega_1) > p(\omega_2)$. The diagonal elements for two classes' covariance matrices for which the half diagonal elements are all zeros are depicted as
$\Sigma_1 d_{ii} = [7, 5, 1, 3, 4, 0.1, 0, 0.5, 0.01, 0.01]$,
$\Sigma_2 d_{ii} = [0.5, 1, 2, 0.9, 0.7, 0, 0.5, 0.1, 0.01, 0.01]$ where $i = 1, 2, ..., 10$. This dataset is used in chapter 8.

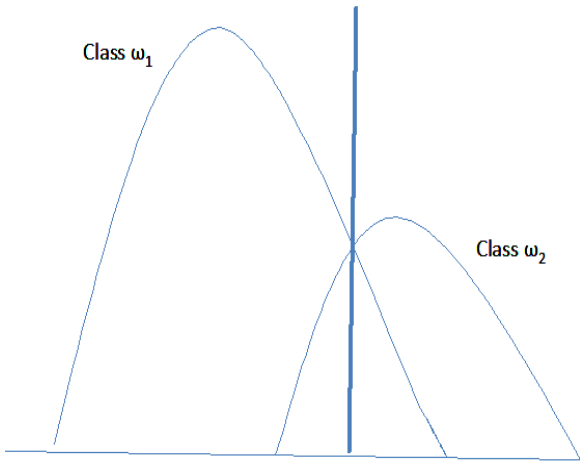


Figure C.3: Data from classes ω_1 and ω_2 where the probability of $p(\omega_1) > p(\omega_2)$.The vertical line indicate class separation.

Appendix D

Description of Other Performance Measures for Unbalanced Data classification

This Appendix, describe the other performance measures that are used for evaluating overall classification performance in case of unbalanced datasets. Further, this thesis also provides a comparative study between the performance measures that are used for unbalanced data based on their individual characteristics.

ROC graph

As $F-measure$ and $G-mean$ output a single scalar value as the performance of a classifier, in order to compare the performance of different classifiers over a range of distributions, Receiver Operating Characteristics (ROC) Curve is widely used by the researchers [43]. ROC curve is graphical representation to depict the trade-off between benefits (TP_{rate}) and loss (FP_{rate}) where the TP_{rate} is plotted on the Y-axis and FP_{rate} is plotted on the X-axis Each classifier produces an (FP_{rate}, TP_{rate}) pair corresponding to a single point in ROC space. Several points in ROC space are important for analyzing the behavior of the classifier. The upper left point (0, 1) represents perfect classification. The lower left point (0, 0) represents the classifier commits only negative classification whereas the opposite strategy the upper right point represents the classifier with only positive classification (1, 1). One point in a ROC diagram dominates another if it is above and to the left. The ROC point falls in the diagonal represents a classifier with random guess over

classification label. The lower right point (0, 1) points to a classifier performance worse than a random guess. If one ROC curve is better than another if it is closer to (0, 1)(TP_{rate} is higher, FP_{rate} lower, or both) and should dominate the other in entire space. In Fig D.1 ROC curve C2 clearly dominates C1. Area under ROC curve (AUC) [97] reduces ROC performance into a single scalar value. Since the AUC is a portion of the area of the unit square, its value will always be between 0 and 1.0. However, because random guessing produces the diagonal line between (0, 0) and (1, 1), which has an area of 0.5, no realistic classifier should have an AUC less than 0.5.

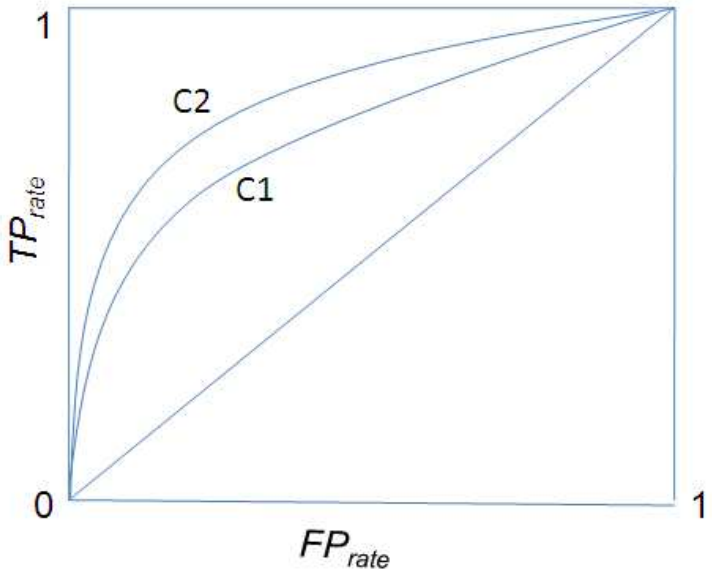
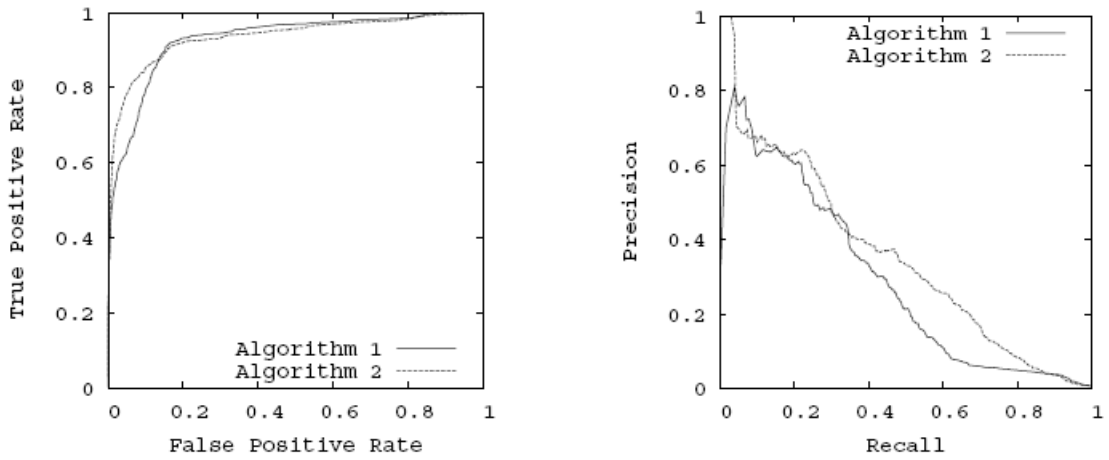


Figure D.1: An ROC graph showing performance for two different classifiers C_1 and C_2

PR curves

In the case of highly skewed data sets, from [31] it is observed that the ROC curve may provide an overly optimistic view of an algorithms performance. Under such situations, the *PR* curves can provide a more informative representation of performance assessment. In case of unbalanced datasets as the majority class samples outnumbers the minority class samples, the drastic change in false positive rates could not be captured by ROC curves due to the large denominator from eq. 3.11. On the hand *precision* whose denominator is the combination of both *TP*

and *FP* from eq. 3.16 could capture the changes in false positives. From Fig D.2a it is clear that, the two algorithms in ROC space and close to optimal whereas the Fig D.2b it is clear that still there is a room for improvement in *PR* space. Therefore from *recall* eq. 3.15 and *precision* eq. 3.16,the *PR* curve is defined by plotting *precision* rate (y-axis) over the *recall* rate (x-axis). However, while the objective of ROC curves is to be in the upper left hand of the ROC space, a dominant *PR*curve resides in the upper right hand of the *PR* space. Further [31] proved that *PR* curves have one-to-one correspondence to ROC curves, further, a curve dominates in ROC space if and only if it dominates in *PR* space. However, an algorithm that optimizes the AUC in the ROC space is not guaranteed to optimize the AUC in *pr* space. In [31] it was also shown that the existence of the *PR* space is analogous to the convex hull in ROC space, which we call an achievable *PR* curve. Hence, *PR* space is an effective evaluation technique, which



(a) Comparison in ROC space

(b) Comparison in PR space

Figure D.2: The difference between comparing algorithms in ROC vs PR space, taken from [31]

has all the characteristics and analogous benefits of ROC.

Cost curves

Another limitation of ROC curves is that they lack the ability to provide insights on a classifiers performance over varying class probabilities or misclassification costs [57, 58]. In order to overcome these limitations cost curves are roposed

[57, 58, 59]. Unlike ROC curves, cost curves are specifically designed for a specific performance measure of expected cost. Cost curves have the ability to explicitly express the classifier's performance in a visual format over a range of class distribution or with varying misclassification cost. The x-axis can represent the "probability cost function "which is a normalized product of $C(-/+)p(+)andC(+/-)p(-)$ and the y-axis represents the "Normalized Expected Cost". Here $C(-/+)$ is the misclassification cost for positive class whereas $C(+/-)$ is the misclassification cost for negative class. In cost curve representation, each (TP, FP) pair in ROC space is transformed to line in cost space by Normalized Expected Cost and PCF(+) (Probability Cost function for the target class). The cost space exhibits mapping for converting the lines in cost space to points in ROC space. Hence it is bi-directional point/line duality between ROC and cost space representation. Any (TP, FP) classification pair in ROC space is related to a line in cost space by

$$E[C] = (1 - TP - FP) * PCF(+) + FP \tag{D.1}$$

$E[C]$ is the expected cost and $PCF(+)$ probability of an example being from positive class. Fig D.3 depicts the mapping of 10 operation points in ROC space to cost space whereas Fig D.4 represents the comparison of two ROC and cost curves. Table D.1 depicts the performance measures and their appropriate usage as a performance measure for unbalanced data classification problem.

Table D.1: Performance measures for two-class classification and their appropriate usage for unbalanced data classification

Measure	Usage
Accuracy	Not appropriate
F-Measure	Single class only (either minority or majority)
G-mean	Whole classifier performnce
ROC	Visual representation, compares different classifiers over range of distributions
PR-curves	Visual representation, holds variations in FP for highly skewed datasets
cost curves	Visual representation for varying misclassification costs

Though, there are several measures that are available to evaluate the classifier performance in case of unbalanced data classification problem, this thesis uses

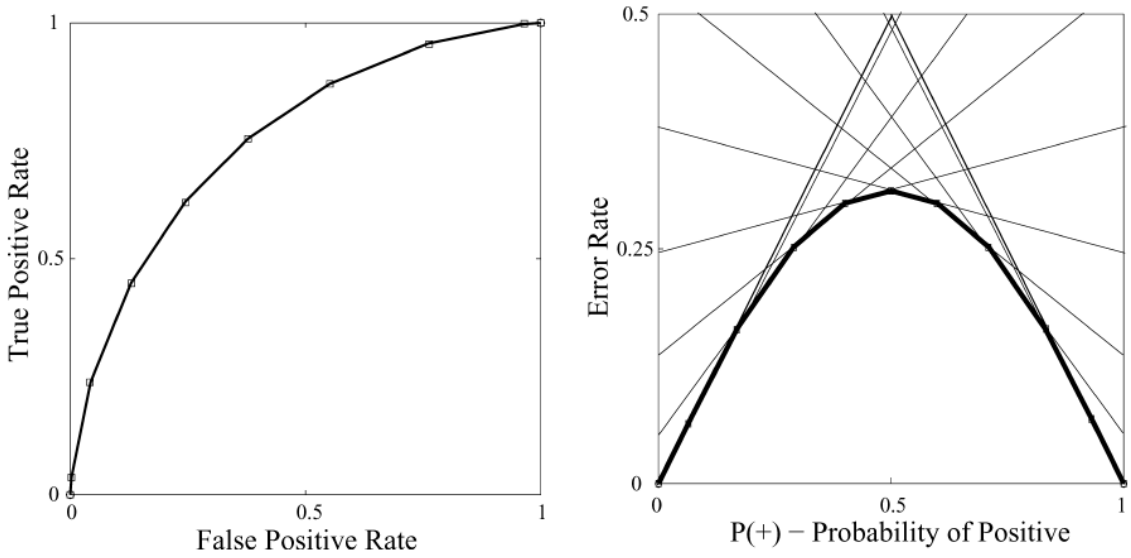


Figure D.3: (a) Ten ROC points and their ROC convex hull (b) Corresponding set of cost lines and their lower envelope, taken from [57]

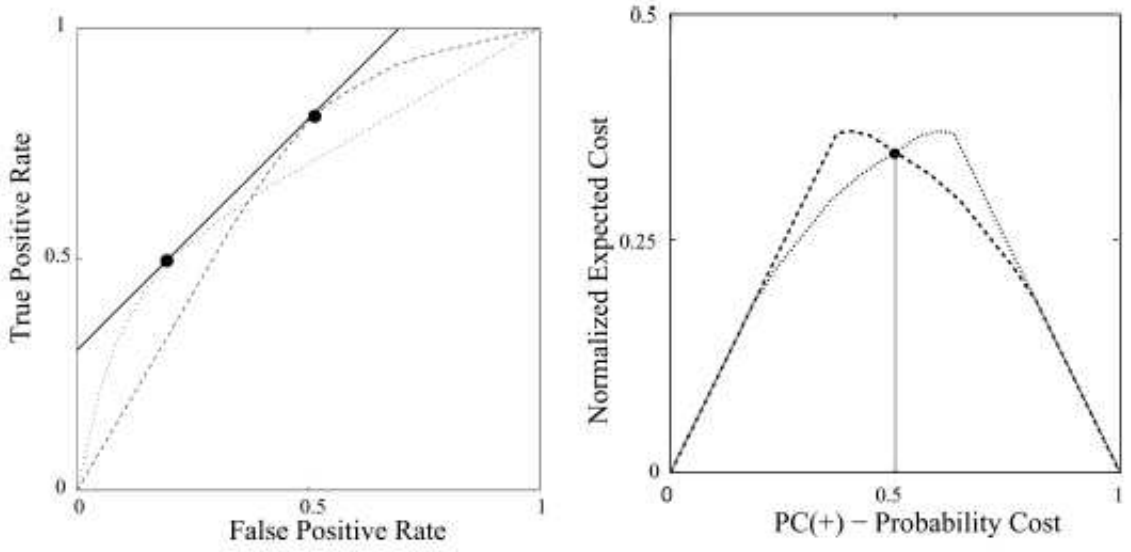


Figure D.4: (a)Two ROC curves that cross (b)Corresponding cost curves [57]

minority class $F-Measure$ and whole classifier's $G-mean$. Minority class $F-Measure$ which reflects the minority class prediction which inherently indicates the trade-off between TP's and FP's. A high minority class $F-Measure$ indicates that both minority as well as majority class predictions are high. Further $G-mean$ was also used in this thesis, on occasions where whole classifier performance was to be evaluated.