RSMG Progress Report 3

(Thesis Proposal)

Class Imbalance Learning

Shuo Wang

Supervisor:

Xin Yao

Thesis Group Members:

Dr. Peter Tino and Dr. Ata Kaban

September 1, 2008

Abstract

Class imbalance learning has recently received considerable attention in machine learning as current algorithms do not provide satisfactory classification performance. Standard algorithms are overwhelmed by majority examples while minority examples contribute very little. A number of improved algorithms have been proposed in the literature, where considerations have been made at the data level and algorithm level.

At the data level, many different forms of re-sampling techniques are proposed, including over-sampling, under-sampling, and combinations of them. They have been shown to achieve good performance on minority examples of two-class data sets. However, the literature has shown little examples of re-sampling used for classifying multi-class data sets. It has been shown that re-sampling techniques tend to degrade when applied to imbalanced data sets with multiple classes. Hence, multi-class classication in imbalanced data sets remains an important topic of research.

At the algorithm level, existing algorithms and techniques are adapted to the special characteristics of imbalanced data sets. Two main groups of learning algorithms are studied: cost-sensitive learning, one-class learning, and classier ensembles. One-class learning has been shown to be useful in cases of extremely unbalanced data sets with high dimensional noisy feature spaces. Ensemble models are one possible improvement on a larger range of data sets, but this technique has been explored in little experimental or theoretical detail.

The thesis will mainly study multi-class classication and the capacity of ensembles used on imbalanced data sets, and how diversity influences the ensemble performance. Current algorithms will be reviewed and analyzed, and new re-sampling strategies and ensemble models will be proposed based on the studies.

Contents

Chapter 1

Introduction

Imbalanced data sets (IDS), also referred to as class imbalance learning, correspond to domains where there are many more instances of some classes than others. Classification on IDS always causes problems because standard machine learning algorithms tend to be overwhelmed by the large classes and ignore the small ones. Most classifiers operate on data drawn from the same distribution as the training data, and assume that maximizing accuracy is the principle goal. Many real-world applications encounter the problem of imbalanced data, such as medical diagnosis, fraud detection, text classification, and oil spills detection. The prediction of minority class is more significant in those cases than majority. Therefore, class imbalance learning draws more and more attention in recent years.

Some solutions to the class imbalance problem have been proposed at both data level and algorithm level. At the data level, various re-sampling techniques are applied to balance class distribution, including over-sampling minority class instances and under-sampling majority class instances. Re-sampling strategies [28, 55, 14] could be categorized into random re-sampling; focused re-sampling, in which part of samples are chosen based on some sampling method rather than randomly selected; and over-sampling with new synthetic data generation. Although they have been showed to achieve success in some applications, over-sampling still confronts overfitting problem and under-sampling has to eliminate useful information potentially.

At the algorithm level, solutions are proposed by adjusting algorithm itself, including adjusting the costs of the various classes to counter the class imbalance (cost-sensitive learning), adjusting the decision threshold or probabilistic estimate, and recognition-based (one-class learning) rather than discrimination-based (two class) learning. Cost-sensitive learning [58, 40, 24, 21, 50] and one-class learning [49, 44, 28, 14, 5] are extended research of class imbalance problem. However, both of them are limited to solve some kinds of applications.

Class imbalance is not the only problem responsible for reducing the performance of learning algorithms. Other factors are identified that hinder classification performance [56, 55, 14, 46, 47], such as the overall size of data sets and concept complexity. Several data conditions are illustrated by some research [33, 35], involving simple

data sets, data sets with overlap, and data sets with small disjuncts. As a part of my research, we will analyze how those imbalance-related factors influence our proposed classification models.

The use of classifier ensembles is a promising technique to improve the performance of weak learners, especially when there is insufficient training data to form a better learning model. A number of different ensemble techniques have been proposed and compared during the past few years. The ensemble procedure of some models provides us a good chance to apply various re-sampling techniques to balance skewed class distribution, such as Bagging and Boosting two of the most popular ensemble techniques. Their algorithm modifies method of forming each bag and is applied to a real-world problem. Furthermore, Negative correlation learning (NCL) is a successful neural network ensemble learning technique developed by Liu, 1998 [41]. It adjusts ensemble diversity explicitly, which can be potentially useful to interpolate diversity among minority examples in imbalances data sets.

Another related issue is multi-class problem. If there is more than one minority class, the situation will be more complex. Some current re-techniques confront degrading trouble. Multiple minority classes make class boundaries hard to be decided, but there are very few studies until now.

The thesis will present study of class imbalance problem, in which two main research questions are proposed. One is multi-class classification. The other is classifier ensembles analysis on imbalance data. Some related topics are involved, including data re-sampling, influence of imbalance-related factors, and ensemble techniques. In order to find out if ensemble approaches give better performance, new Bagging variations are proposed and tested on artificial data sets at first. We are also interested in properties of NCL and Boosting to classify imbalanced data set. They are discussed in more detail in the following sections.

The remainder of this section introduces the problems addressed in this work and our motivations. The first subsection defines several important concepts and problems involved in our research: imbalanced data sets (IDS), multi-class problem, and classifier ensembles. The second subsection describes the motivations for proposed problems. The third subsection addresses my research questions clearly. The forth subsection gives the structure of this thesis proposal. The final subsection introduces several important notations used in the proposal.

1.1 Problem Overview

In this section, we give several concepts and definitions related to imbalanced data sets.

Imbalanced data sets (IDS) problem, also called class imbalance problem, commonly corresponds to the problem encountered by inductive learning algorithms on domains for which some classes are represented by a large number of instances while others

are represented by only a few. We normally meet two-class problems, which means one class has much more instances than the other. Sometimes, we also have multi-class cases, in which there are not enough instances for more than one class. It may cause more trouble when deciding classification boundary. Since 2000, several issues have been under discussion [14]:

- Clever re-sampling and combination methods;
- How to evaluate learning algorithms in the case of class imbalance;
- The relationship between class imbalance and cost-sensitive learning;

Re-sampling is a group of techniques by managing training data sets to change imbalanced distribution. A number of related methods are discussed. They are data level solutions. There is no need to change algorithms themselves. At algorithm level, fewer solutions are proposed. Two main learning topics are studied: one-class learning and cost-sensitive learning. *One-class learning* is a category of algorithms for solving imbalanced or high noisy data sets, learning from one class rather than discriminating two classes. *Cost-sensitive learning* can be seen as a particular topic in IDS, in which case cost matrix is needed for different types of errors or examples during classification. Class penalty is given depending on the defined cost. Minority class examples have higher cost value if they are misclassified. However, we cannot often get the cost matrix. To some extent, re-sampling techniques are more flexible and simpler. Popular techniques on IDS will be described in Section 2 "Literature Review".

As a part of my research, ensemble is chosen as a potentially effective way to solve class imbalance problem. Ensemble is defined as follows [17, 6]:"*An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples.*" Each algorithm takes an learner and a training set as input and runs the learner multiple times by changing the distribution of training set instances. The generated classifiers are then combined to create a final classifier that is used to classify the test set. Base learner (or base inducer, ensemble member) could be decision trees, Naive-Bayes, Neural Network, etc. We use the term base learner to refer to learning algorithm used for constructing each classifier. We use the term ensemble member directly when discussing about each ensemble committee member. Classifier combination methods, such as Bagging and Boosting, have been shown to be very successful in improving accuracy for artificial and real-world datasets. Section 2 will give a review on ensemble techniques.

1.2 Importance of Research and Aims of the Thesis

As mentioned earlier, class imbalance problem is prevalent in many practical applications, including: medical diagnosis, fraud detection, text classification, risk management, etc. More and more researchers realized that their data sets were imbalanced, and that lack of information caused suboptimal classification performance or even worse. Insufficient information is intrinsic in some domains. For example, within a given data set, there are typically very few cases of cancer as compared to the large number of non-cancer patients. There are also non-intrinsic domains, which means it is hard to collect data due to some reasons, such as privacy and expenditure. Both cases tell us it is worthwhile spending more time in analyzing minority data. Therefore, the fact is that we need to analyze limited information from minority and other information from majority as much as possible.

What is wrong with current machine learning algorithms? [55] First reason is accuracy. Standard algorithms are driven by accuracy and try to discriminate difference among different classes, in which case minority data is always ignored. Second reason is class distribution. The current classifiers assume that the algorithms will operate on data drawn from the same distribution as the training data. Third reason is error costs. The current classifiers assume that the errors coming from different classes have the same costs. Forth reason is data structure. It is assumed that training data is not much different from the data to test. This is not always true in some cases that may contain heterogeneous data.

With regard to the first issue, clever improved or new learning methods are explored from both data level and algorithm level. Though all these studies shed some light on the way various methods compare, there is no single final word on the question. In other words, a number of techniques were shown to be effective if applied in a certain context, where the breadth of the context may vary. More work put their attention on various re-sampling methods. That brings us to question: Is sampling becoming a *de facto* standard for countering imbalance? In addition, when classifying multi-class data sets, some current learning models may suffer problems, especially for data-level techniques. For example, if there are two minority classes, how should re-sampling rate be decided? If we do not use random sampling, how should we choose which examples we want to eliminate or use to generate new examples. However, there are still not many researches working on this topic yet.

With regard to the second issue, class distribution is an important issue for learning. The training data might be imbalanced but the testing might not and the other way around. However, experimental studies show that a balanced class distribution is not the best for learning (Weiss & Provost 2003) [57], (Visa & Ralescu 2005) [54] and the question is: What is the best class distribution for learning a given task?

With regard to the third issue, the error costs are different in most applications. If the error costs and class distribution are known the correct classification threshold can be computed. But the difficulty is that error costs are hard to assess even by the

human experts in the field, and therefore, these costs are rarely known. Further, it is important to mention that, classifiers have problems even for the balanced data, when the errors coming from different classes have different but unknown cost.

The forth issue is a particular case in imbalanced data sets. No studies directly consider heterogeneous data in IDS. Heterogeneity could happen between training data and testing data. It causes the following problems: Is there a transductive learning solution for heterogeneous data? However, when looking through this problem, we find it is hard to give an accurate problem formulation or get a proper data like this. So, we do not consider this issue in the thesis.

Besides, some papers discussed interaction between the class imbalance and other issues such as size of data set and concept complexity. It was found that those imbalance-related factors encumber classification performance in certain cases. It is worth noting that analysis of those factors is helpful to solve imbalance class problem. Finally, there is too much reliance of the class-imbalance research on C4.5 [14]. It was argued that the community should focus on it less. We need to explore solutions from a new angle.

According to the above discussion and current issues, the thesis is aiming at the following points:

- Study re-sampling techniques on imbalance problem for both two-class and multi-class classification.
- How to use ensemble models deal with imbalance problem from both data level and algorithm level, including diversity analysis.
- Study the influence of imbalance-related factors on proposed solutions.

1.3 Motivations

Motivations of my research are presented in this section. Due to challenge to predict minority class instances accurately, it is worthwhile to study how ensemble models can contribute to imbalanced data sets. Reasons of using ensemble models are explained firstly.

Over the past few years, ensembles have emerged as a promising technique with the ability to improve the performance of weak classification algorithms. It has been discovered that an ensemble of classifiers can often outperform a single classifier. There exist a number of research on how to form a classifier ensemble and why ensemble techniques work [6, 18, 19, 45, 16, 53]. From the stand of ensemble itself, multiple classifiers can average uncertainty of each classifier especially when the training sample is small, and reduce the risk of choosing the wrong classifier. Besides, ensemble techniques are flexible to use many kinds of learning algorithms as base learner,

aiming at different practical applications. From the stand of imbalanced data distribution, re-sampling techniques, common techniques for balance class distribution, can be easily combined with ensemble models for drawing random sets of the available data in the absence of adequate training data. For example, Bagging and Boosting both have re-sampling procedure to choose data subsets. Their advantage over other methods is that they are external and easily transportable. "External" means algorithm-independent or not algorithm-specific. Therefore, ensemble model could be a good choice for imbalance. After identifying imbalance-related factors, it is also interesting to study how those factors influence the performance of ensemble techniques.

There is always a trade-off between simplicity and accuracy. Re-sampling is common and efficient to imbalance problem because it is simple and algorithm-independent. There are a number of related techniques proposed to solve specific problem [5, 12, 23, 4]. Algorithm level methods have their problem scope that can be solved, but they provide more accurate result. It is hard to judge which method is definitely better than others. They outperform when they are used in a proper context. When looking for a breakthrough, I am considering using negative correlation learning (NCL) to improve classification accuracy on minority class further but not to sacrifice overall performance. NCL is a successful neural network ensemble model, which encourage diversity among ensemble members directly. It has been showed to outperform other ensemble models (Islam et al.; 2003; Wang et al.; 2004; Chandra and Yao; 2006). Considering diversity is an important aspect for final performance of ensemble models, it is possible that NCL is potentially helpful to improve accuracy on minority class by increasing its classification diversity. Therefore, it is worthwhile to explore the new way.

1.4 Structure of Proposal

The thesis proposal is organized as follows.

Chapter 1 presents main research topics and problem overview briefly. It explains why the work is important and our motivations on this work.

Chapter 2 introduces the knowledge background essential to understanding this domain of work. Three topics are reviewed: class imbalance problem, ensembles of learning machines, and ensemble models in imbalanced data sets.

Chapter 3 analyzes inadequacies in previous work and formulates research questions. We give formal definitions of those questions.

Chapter 4 describes what I have done so far, an empirical study of bagging variations for learning from imbalanced data sets. We generate imbalanced data sets to control imbalance-related factors and introduce ensemble decision tree model following by some experimental results.

Chapte 5 proposes several research ideas and plans for my next step study and explains why the proposed work is important and how we carry on current research.

Chapter 6 discusses how the work should be evaluated and what is the contribution, and Chapter 7 gives the timetable for the next two years' study.

1.5 Notation

This section lists some important notations used in the following sections.

x – vector of input data with d dimensions, domain $\mathbb{R}^d$.

y – the expected output corresponding to a given input x.

N – the number of training examples.

M – the number of individual models, i.e. the size of ensemble

$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ – the training data set.

$\mathcal{D}_i(\cdot)$ – the i-th subsets from data set $\mathcal{D}$

$\bar{f}(\cdot)$ – the ensemble function.

$f_i(\cdot)$ – the i-th individual model function in an ensemble.

Chapter 2

Literature Review

This chapter reviews some concepts and learning approaches on class imbalance problem, and their advantages and disadvantages. Ensemble learning models are also introduced as one part of our research.

2.1 Class Imbalance Learning

A number of solutions to the class-imbalance problem were previously proposed both at the data and algorithmic levels [14]. At the data level, these solutions include many different forms of re-sampling such as random over-sampling with replacement, random under-sampling, focused (or directed) over-sampling where no new examples are created, but the choice of samples to replace is focused rather than random, focused (or directed) under-sampling where the choice of examples to eliminate is focused, over-sampling by generating new samples, and combinations of the above techniques [5, 12, 23, 4]. At the algorithmic level, solutions include adjusting the costs of the various classes so as to counter the class imbalance (cost-sensitive learning) [58, 40, 24, 21, 50], adjusting the probabilistic estimate at the tree leaf (when working with decision trees), adjusting the decision threshold, and recognition-based (one-class learning) [34, 49] rather than discrimination-based (two class) learning. Some important solutions presented in the workshops are discussed in the thesis.

2.1.1 Re-sampling

The basic sampling methods include under-sampling and over-sampling. Under-sampling eliminates majority-class examples while over-sampling increases the number of minority-class examples. Both of these sampling techniques decrease the overall level of class imbalance, thereby making the rare class less rare. These sampling methods have several drawbacks with no doubt. Under-sampling discards

potentially useful majority-class examples and thus can degrade classifier performance. Therefore, they are normally used on very large data sets in which there are enough redundant data to be removed. Over-sampling, however, can increase the time necessary to build a classifier because it introduces additional training cases. Even worse, because over-sampling often involves making exact copies of examples, it can lead to overfitting [13, 23]. As an extreme case, final classification rules may be introduced to cover a single, replicated, example. More importantly, over-sampling introduces no new data, so it does not address the fundamental "lack of data" issue.

Some clever sampling methods are proposed which may use instance selection algorithms when removing or adding examples or combine under-sampling and over-sampling techniques. Some studies have recognized the fact that it is still unclear which sampling method performs best, what sampling rate should be used, and that the proper choice is probably domain specific. Common techniques are reviewed in the following:

Over-sampling

Random Over-sampling: Balance class distribution by replicating minority class examples randomly, but it increases the likelihood of overfitting since it makes exact copies. Make decision regions of the learner more specific and closer to minority class.

SMOTE [13]: Generate new synthetic minority examples by interpolating between minority examples that lie together. It makes the decision regions larger towards majority class and less specific. Synthetic examples are introduced along the line segment between each minority class example and one of its k minority class nearest neighbors. Its generation procedure for each minority class example can be explained as: firstly, choose one of its k minority class nearest neighbors. Then, take the difference between the two vectors. Finally, multiply the difference by a random number of 0 or 1, and add it to this example.

Borderline-SMOTE1, borderline-SMOTE2 [30]: Based on SMOTE algorithm, they generate new minority examples by using the examples only close to decision boundary, and achieve better TP rate and F-value. They considered that examples on the borderline and the ones nearby are more important for classification. First, they find out the borderline minority examples by calculating k nearest neighbors. Then SMOTE is applied only to those borderline minority examples to generate new examples. The difference between borderline-SMOTE1 and borderline-SMOTE2 is that the first method generates synthetic examples from original minority examples and their minority nearest neighbors while the other method also generates synthetic examples from their majority nearest neighbors besides minority ones. They experimentally prove that borderline-SMOTE1 behaves good on both recall value and F-value, and borderline-SMOTE2 achieves even better recall value but F-value is decreased because overlap is caused between two classes.

Under-sampling

Random Under-sampling: Balance class distribution by removing majority class examples randomly. Main disadvantage is that it discards data that may contain useful information

Condensed Nearest Neighbor Rule [31]: Condensed Nearest Neighbor Rule (CNN)) is used to find a consistent subset of examples by Hart in 1968. Condensed Nearest Neighbor Rule (CNN) is used to find a consistent subset of examples by Hart in 1968. A subset $\hat{E} \subseteq E$ is consistent with E if using a 1-nearest neighbor (1-NN), $\hat{E}$ correctly classifies the examples in E. An algorithm to create a subset $\hat{E}$ from E as an under-sampling method is the following: Firstly, randomly draw one majority class example and all examples from the minority class and put these examples in $\hat{E}$. Afterwards, use a 1-NN over the examples in $\hat{E}$ to classify the examples in E. Every misclassified example from E is moved to $\hat{E}$. This procedure does not guarantee that it finds the smallest consistent subset from E, but the idea is to eliminate the examples from the majority class that are distant from the decision border, since these sorts of examples might be considered redundant for learning.

Tomek Links [51]: Tomek links was proposed as a data cleaning method in 1976 by Tomek. Given two examples x_i and x_j belonging to different classes, and $d(x_i, x_j)$ is the distance between x_i and x_j. A (x_i, x_j) pair is called a Tomek link if there is not an example x_k, such that $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_i, x_j)$. If two examples form a Tomek link, then either one of these examples is noise or both examples are borderline. When it is used as an under-sampling method, only examples belonging to the majority class are eliminated, so as to expand borderline towards majority class.

One-sided Selection [38]: The idea of one-sided selection (OSS) is to remove both redundant and borderline examples from majority class, and keep "safe" examples for classification. Firstly, redundant examples are removed by using CNN algorithm. Then, borderline examples are followed that participate at Tomek Links.

Neighborhood Cleaning Rule [36]: Neighborhood cleaning rule uses kNN to find borderline examples of majority class and remove them. For a two-class problem, for each majority example x_i in the training set, its three nearest neighbors are found. If two or more of them are minority examples, then x_i is removed. That means the class of x_i contradicts the class given by its nearest neighbors. If x_i belongs to minority class and its nearest neighbors misclassify x_i, then the nearest neighbors that belong to majority class are removed. A decontamination methodology is also based on this idea on imbalanced data sets [2].

Evolutionary Prototype Selection [27]: Evolutionary prototype selection (EPS) uses genetic algorithm to evolve subsets of original training data sets and find a best subset for classification. In the imbalanced context, this is used for under-sampling majority class.

The above discussion of under-sampling techniques shows that their objective is

to remove two types of examples from majority class - redundant examples and borderline examples (which are regarded as unreliable). Its direct result is forcing classification boundary moving towards majority class, and hence avoiding overfitting minority class. It is also noted that combining focused over and under-sampling, such as SMOTE+Tomek is applicable when the data sets are highly imbalanced or there are very few instances of the minority class.

2.1.2 One-class Learning (Recognition-based Approach)

One-class learning is also called recognition-based learning. As described by its name, one-class learning approach provides an alternative way to do classification where the model is created based on one-class examples alone. It is assumed that only one class information is available and no information about the other class is present. The boundary between two classes has to be estimated from data of the only one class, the target class. Its general idea avoids the essential problem caused by imbalanced data set, which is performance of discriminative approaches is overwhelmed by majority class examples and minority class examples tend to be ignored. Mainly, two kinds of learning algorithms were studied in the context of one-class approach for imbalanced data sets the boundary methods (SVMs, etc.), and the reconstruction methods (autoencoders, etc.).

Boundary Methods

A closed boundary around the target data set is optimized in boundary methods. In most cases, distances or weighted distances to a set of examples in the training data set are computed. The threshold on the output is then obtained in a direct way. The boundary methods rely on the distances between examples heavily.

Reconstruction Methods

Reconstruction methods are to represent inputs of target class into another form in which common information contained in target class is trying to be kept. It is assumed that outlier examples (examples which is not in target class) do not satisfy the target distribution. They should be represented worse than true target examples and their reconstruction error should be high. Autoencoders [34] is one of the reconstruction methods used to solve imbalance problem. It is trained to reproduce the input patterns at their output layer, aiming at the target examples could be reconstructed with smaller error than non-target examples.

Under certain conditions, one class approaches to solving the classification problem may in fact be superior to discriminative (two-class) approaches (such as decision trees or Neural Networks). In particular, [48] shows that one-class learning is particularly useful when dealing with extremely unbalanced data sets composed and high dimensional noisy feature space.

Table 2.1: Confusion Matrix for Two-class

	Positive prediction	Negative prediction
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

2.1.3 Cost-sensitive Learning

Unlike traditional learning models, cost-sensitive learning uses misclassification costs to balance the difference among classes of training data sets. In general, misclassification costs are described by cost matrix U with $U(i, j)$ being the cost of predicting an example belongs to class i when in fact it belongs to class j. The goal of cost-sensitive learning is to minimize the cost of misclassification. In class imbalance problem, the cost of misclassifying a minority class example is usually more expensive than that of misclassifying a majority class example. Thus, cost-sensitive learning is studied as a solution of class imbalance learning. Liu [40] gave an experimental analysis on how class imbalance affects cost-sensitive learning methods and suggested that not-seriously imbalanced data sets can be solved by simply applying cost-sensitive learning methods. When we deal with a seriously imbalanced data set, it is better to balance the class distribution first before applying cost-sensitive learning methods.

2.1.4 Evaluation Metrics for Class Imbalance Learning

As pointed out by many authors [3], the performance of a classifier in applications with class imbalance must not be expressed in terms of the average accuracy. For instance, consider a domain where only 2% examples are positive. In such a situation, labeling all new samples as negative would give an accuracy of 98%, but failing on all positive cases. Consequently, in environments with imbalanced classes, additional measures have been proposed. The most common metric is ROC analysis and the associated use of the area under the ROC curve (AUC) to assess overall classification performance. The geometric mean (G-mean) is another good indicator on overall performance. For measuring performance of one class, precision and recall are metrics by considering only one class (minority or majority) that are useful for data mining. One variation of precision and recall is F-measure. The F-measure is parameterized and can be adjusted to specify the relative importance between precision and recall. First, we define the confusion matrix for a two-class problem in Table 2.1. Then, the following part will give a detailed description and definition.

From this table, four simple measures can be directly obtained: TP and TN denote the number of positive and negative examples classified correctly, while FP and FN denote the number of positive and negative examples misclassified respectively. In the following parts, positive represents minority and negative represents majority by default.

1. True positive rate: $TPR = TP/(TP + FN)$.
   It is the percentage of positive examples correctly classified within positive class, also referred to as recall or sensitivity.

   True negative rate: $TNR = TN/(FP + TN)$.
   It is the percentage of negative examples correctly classified within negative class, also referred to as specificity.

   False positive rate: $FPR = FP/(FP + TN)$.
   It is the percentage of negative examples misclassified as belonging to the positive class.

   False negative rate: $FNR = FN/(TP + FN)$.
   It is the percentage of positive examples misclassified as belonging to the negative class.

2. Precision: $precision = TP/(TP + FP)$
   It is the proportion of positive examples that are actually positive, representing how accurate the learning model is.

3. F-measure
   As another one class metric, F-measure is defined as:
   $F - value = \frac{(1+\beta^2)\cdot recall \cdot precision}{\beta^2 \cdot recall + precision}$, where β corresponds to relative importance of precision and recall, and it is usually set to 1. Value of F-measure (or F-value) incorporates both precision and recall, in order to measure the "goodness" of a learning algorithm for the class.

4. G-mean
   G-mean is an overall performance metric and defined as:
   $\sqrt{positiveAcc \cdot negativeAcc}$, where positiveAcc and negativeAcc are TP rate and TN rate respectively. This measure relates to a point on the ROC curve and the idea is to maximize the accuracy on each of the two classes in order to balance both classes at the same time.

5. ROC curve
   ROC curve is a two-dimensional graph to select possibly optimal models based on the TP rate and FP rate. It also represents trade-offs between benefits (TP) and costs (FP). In the ROC curve, the TP rate is represented on the Y-axis and the FP rate on the X-axis. Each prediction result or one instance of a confusion matrix represents one point in the ROC space. Several points on a ROC graph should be noted. The lower left point $(0, 0)$ represents that the classifier labeled all examples as negative. The upper right point $(1, 1)$ is the case where all examples are classified as positive. The point $(0, 1)$ represents perfect classification, and the line $y = x$ defines the strategy of randomly guessing the class.

   In order for assessing the overall performance of a classifier, one can measure the fraction of the total area that falls under the ROC curve (AUC). AUC varies between 0 and +1. Larger AUC values indicate generally better classifier performance.

2.2 Classifier Ensembles

The key idea in ensemble learning is that a committee of classifiers will produce better results than a single classifier in terms of stability and accuracy in many situations. This is particularly the case for weak learners, such as neural networks and decision trees. The *Condorcet's Jury Theorem* states that:

If each voter has a probability p of being correct and the probability of a majority of voters being correct is P, then $p > 0.5$ *implies* $P > p$*. In the limit,* P *approaches* 1*, for all* $p > 0.5$*, as the number of voters approaches infinity. On the other hand, if p is less than* 0.5*, then adding more voters make things worse.*

This theorem was proposed by the Marquis of Condorcet in 1784 [16]. Therefore, ensemble performance P will be greater than p only if there is diversity in the pool of voters. And accuracy of each voter p is the other key factor, which must have better prediction than 0.5 at least. The probability of the ensemble being correct will only increase as the ensemble grows if the diversity in the ensemble continues to grow as well. Krogh and Vedelsby [37] have shown that the reduction in error due to an ensemble is directly proportionate to the diversity or ambiguity in the predictions of the components of the ensemble as measured by variance. Gavin [10] also gave a clear explanation of ensemble diversity by presenting several important decompositions of mean squared error (MSE) for regression problems. It is difficult to show such a direct relationship for classification tasks but it is clear that the uplift due to the ensemble depends on the diversity in the ensemble members and accuracy of each member.

2.2.1 Ensemble of Learning Machines

There are a number of ensemble models and various variations proposed to solve real-world applications, such as Bagging and Boosting. They have been shown to be very successful in improving the accuracy of certain classifiers.We review the most important approaches - Bagging and AdaBoost in this section, and give a conclusion of popular ones that many applications use in the table 2.2.

Bagging

The Bagging algorithm, the acronym of bootstrap aggregating, is proposed by Breiman in 1996 [7]. Each member classifier is generated by different bootstrap samples, in which each bootstrap sample is drawn uniformly from original training data set at random with replacement. That means it allows some examples may appear multiple times while some may not be chosen at all. When a new example comes, each classifier will give a vote and choose the highest one as final output. Each training example has probability 63.2% of being selected at least once on average. Classifier diversity is ensured implicitly by the bootstrap procedure. In addition, a

relatively unstable base learner is recommended so as to obtain sufficiently different decision boundaries for small perturbations in different training subsets, such as decision trees and neural networks.

Boosting

Boosting was introduced by Schapire (1990) as a method for boosting the performance of a weak learning algorithm. AdaBoost (Adaptive Boosting) was introduced by Freund & Schapire [26] and drawed a remarkable attention, because it is more commonly used and is capable of handling multi-class and regression problems. The AdaBoost algorithm generates a set of classifiers by re-sampling like Bagging, but the two algorithms differ substantially. The AdaBoost algorithm generates the classifiers sequentially, while Bagging can generate them in parallel. Each training example keeps a weight in AdaBoost and is updated after each time of iteration of constructing a classifier. The examples which are misclassified currently will be assigned larger weight, in order to be more likely to be chosen as a member of training subset during re-sampling at next round. Hence, consecutive classifiers tend to focus on "hard" examples. A final classifier is formed using a weighted voting scheme-the weight of each classifier depends on its performance on the training set used to build it. Although Boosting has achieved great success in machine learning field, many studies have found it is not resilient to noise, which means it has overfitting problem in noisy cases. Diversity in Boosting methods is ensured explicitly by updating distribution of training data after each time of iteration.

Conclusion of Commonly Used Ensemble Methods

Here we conclude some commonly used ensemble methods in a table and categorize them briefly based on their construction steps.

Table 2.2: Ensemble Methods

Ensemble Method	Description	Category
Bagging [7] (Breiman)	Bootstrap aggregation technique	Manipulate training examples. Each classifier is independent of the others.
AdaBoost [26] (Freund and Schapire)	Maintain a set of weights over the training items, and place more weight on items misclassified by classifier.	Manipulate training examples. Construct each classifier sequentially.
Error-correcting output coding [17, 19] (Dietterich)	If the number of classes, K, is large, randomly partition the K classes into two subsets, re-label the original classes, and construct a classifier.	Manipulate output targets. Each classifier is independent of the others.

Randomized C4.5 / Randomization [18] (Dietterich)	At each node in decision tree, the 20 best tests are determined and one of them is randomly selected for use at that node.	Injecting randomness to feature selection. Each classifier is independent of the others.
Random Subspaces [32] (Ho)	Select random subsets of the available features to be used in training the individual classifiers in an ensemble.	Injecting randomness to feature selection. Each classifier is independent of the others.
Random Forests [9] (Breiman)	Random Subspaces + bagging	Each classifier is independent of the others.

In the above methods, Bagging, AdaBoost, and randomization are mostly used as classification solution. Some research [1, 17, 6, 19] has carried out comparison among these three techniques. Those experiments show that Adaboost often gives the best results in low-noise case. Bagging and randomization give similar performance. Adaboost tends to overfit data while Bagging is shown to work best in the presence of noise.

2.2.2 Bias-Variance Decomposition and Diversity Analysis

Several authors [8, 42, 20] have been working on theories for the effectiveness of ensembles based on bias plus variance decomposition of classification error. This decomposition presents the expected error of a learning algorithm on a particular target function and training set size with three components:

- Intrinsic "target noise": This noise is inherent in the learning problem and is effectively a lower bound on the error that can be achieved by the classifier. It reflects shortcomings in the potential of the available features to capture the phenomenon, and we cannot do anything about it.
- Bias: This captures how the average prediction of the learning algorithm matches the target.
- Variance: This quantifies how much the learning algorithm "bounces around" for the difference training sets of a given size.

For ensemble models, the error function can be extended further to bias-variance-covariance decomposition by Ueda and Nakano in 1996 [52]. It illustrates the generalization error of an ensemble also depends on the covariance between the classifier members. Krogh and Vedelsby [37] give the ambiguity decomposition at a single

data of the ensemble model in 1995. Literature [10] bridges the connection between these two decompositions, and gives a systematic and theoretic analysis about how to explicitly control diversity through the error function and the relationship between the error function and negative correlation learning (NCL) method. NCL is a neural network ensemble model, which manages diversity explicitly by adding a penalty term in its error function. It will be introduced in section 5.2.

2.3 Ensemble Models in Class Imbalance Learning

BEV

A Bagging ensemble variation (BEV) [39] is proposed to classify railroad wheel inspection data, which is highly imbalanced, by Cen Li in 2007. Instead of randomly sampling the original training data with replacements, BEV constructs each bag by containing all minority data and same size of majority data to make each "bag" for training balanced. Assuming that the classifiers are learned from M different sets of training data, BEV system divides majority class data into $M = \lfloor N_{maj}/N_{min} \rfloor$ disjoint sets, where the N_{maj} is the number of majority class data and N_{min} is the number of minority class data. The final result is the majority vote from the M classifiers. Essentially, each "bag" under-samples majority class instances with no replacement and keeps all minority class instances.

SMOTEBoost

SMOTEBoost algorithm [15] combines the Synthetic Minority Oversampling Technique (SMOTE) and the standard boosting procedure.It utilizes SMOTE [13] for improving the accuracy over the minority classes and utilizes boosting to not sacrifice accuracy over the entire data set. This algorithm recognizes that Boosting may suffer from the same problems as over-sampling, such as overfitting, since Boosting tends to weight examples belonging to the minority classes more than those belonging to the common classes. Instead of changing the distribution of training data by updating the weights associated with each example, SMOTEBoost alters the distribution by adding new minority-class examples using the SMOTE algorithm. Experimental results indicate that this approach allows SMOTEBoost to achieve higher F-values than standard Boosting and SMOTE algorithm with a single classifier.

AdaCost

AdaCost [25] is another variant of Boosting method used to discriminate classes with unequal misclassification costs. Different with updating rule in AdaBoost, AdaCost

introduces a misclassification cost adjustment function into its weight updating formula. It increases the weights of false negatives more than false positives. It also decreases the weights of true positives more than true negatives. Under this updating rule, examples will get higher weights and less expensive examples will get lower weights. Hence, the final voted ensemble would predict more costly instances correctly since each weak hypothesis predicts more expensive examples correctly for such a distribution. In addition, it also gives an upper bound on the cumulative misclassification cost.

DataBoost

Different from previous solutions of imbalanced data, DataBoost [29] not only tries to improve the predictive accuracies of minority class, but also that of majority class. In other words, it combines data generation and boosting procedures to improve performance of minority class without sacrificing the performance of majority class. During the execution of Boosting, hard examples from both majority and minority classes are identified. Then they are used to generate synthetic examples separately. The synthetic examples will be added to the original training data set to balance class distribution and update example weights. The class frequencies in the new training set are rebalanced to alleviate the learning algorithms bias toward the majority class. In this way, the author focuses on improving the predictions of both the minority and majority classes.

Lazy Bagging

Lazy Bagging (LB) [59] is "lazy" because it will not build bootstrap replicate bags until a test example comes. Then it uses the characteristic of the test example by considering the distances with other training data (kNN) and generates a Bagging model suitable to this test example. They assume that an unlabeled example's nearest neighbors provide valuable information for learners to refine their local decision boundaries for classifying this example. This strategy is beneficial for classifying imbalanced data because refining local decision boundaries can help a learner reduce its inherent bias towards the majority class and improve its performance on minority class examples. However, the determination of value k, number of neighbors, is a big problem, especially when the data set is highly skewed.

Chapter 3

Research Questions

This section describes main problems existing in class imbalance learning, and formulates those problems in a clear way.

3.1 Inadequacies in Previous Work

Certain problems have emerged from reviewing the literature. Firstly multi-class classification needs further research in class imbalance cases from both data level and algorithm level. It causes performance degrading by introducing complexity of data. Although there are a few solutions that have been proposed aiming at multi-class, most current algorithms are discussed and tested by using two-class imbalanced data sets. From data level, re-sampling is the main way by balancing class distribution. In two-class, most re-sampling techniques are trying to manage boundary between majority class and minority class by expanding boundary to majority and giving more room to minority. If extended to multi-class, particularly if there are more than one minority classes, some of those techniques may less effective or worse even. Some approaches from algorithm level have the same problem. Therefore, we need to find out which techniques will degrade greatly or not in common imbalanced data sets with multiple classes firstly. And then we will explore the methods to overcome this problem and make them more robust. For example, if we use re-sampling methods, the first problem we meet is that how we should set over-sampling rate of minority class or under-smapling rate of majority class. It is worthwhile to do more research on re-sampling techniques, because it is independent with specific algorithm or particular application. In addition, some model evaluation metrics for measuring overall performance are not suitable to multi-class cases, such as ROC and G-mean. It is also a subtopic of our research to work on.

Secondly, there is still room for novel ways of classifying imbalanced data sets from algorithm level. Classifier ensemble is a good model for classifying imbalanced data but still comparatively new to this topic. Therefore, our research will put more focus on ensemble models, propose new scheme to solve imbalance problem, and give

comprehensive analysis including diversity analysis and execution time. Diversity analysis will be an important part in our research due to its impact on ensemble models.

3.2 Research Questions and Problem Formulation

There are two main research questions in our study of imbalanced data multi-class classification and ensemble model analysis. Each of them includes several subtopics we take into consideration.

Question 1. Classifying imbalanced data sets with multiple classes

How do current algorithms perform when given imbalanced data sets with multiple minority classes (including both data-level methods and algorithm-level methods)? If not, how to improve current approaches, or explore a new way to solve this problem?

Given:

- Artificial or real-world data sets, containing two or more minority classes.
- Learning algorithms (re-sampling techniques, cost-sensitive learning, one-class learning, ensemble models, etc.).
- Selected evaluation criteria (ROC, recall value, F-measure, etc.).

Research questions:

- Compare performance among different algorithms and the case of single minority class.
- What kinds of algorithms have no multi-class problem? If performance degrades due to multi-class, what are the weaknesses of current algorithms when classifying multi-class data sets?
- How to overcome those disadvantages? Is there a new way or improved way to solve multi-class problem?
- For re-sampling techniques, how to re-define re-sampling rate?
- How to measure overall performance of multi-class? In other words, how to expand two-class evaluation metrics to multi-class context?

Question 2. Performance of ensemble models and diversity analysis

It involves three subtopics - analysis of current ensemble models, propose new ensemble model, and diversity analysis.

Given:

- Artificial data: different settings of imbalance rate, size of data set, and concept complexity in training data set.
  Real-world data sets.
- Ensemble models (Bagging, Boosting, NCL, etc.).
- Base learners (Decision Tree, Neural Network, etc.).
- Selected evaluation criteria (ROC, recall value, F-measure, etc.).

Research questions:

- Evaluate performance of both minority and majority class from different evaluation criteria. The comparison is among different ensemble models, or between ensemble and other solutions (such as cost-sensitive or one-class learning).
- What ensemble models and their variations could improve the classification of imbalanced data? Under what kind of condition, which learning model gives better results?
- How to combine re-sampling techniques with ensemble models?
- Diversity analysis. Which diversity measurements provide more information when classifying imbalanced data? How does diversity influence the prediction of minority data? Can we make use of diversity information to improve algorithms?

Chapter 4

Work Done so far

This section presents some works done during the past few months. First subsection gives an overview of our work and the objective in this part. Second subsection presents two data generation models used in our experiments. The rest subsections explain our algorithms and experimental designs.

4.1 Work Overview and Objective

In this work, we are interested in studying ensemble performance on class imbalance learning with different re-sampling techniques. New Bagging variations are tested and compared. We also try to find out how imbalance-related factors affect the final result by generating corresponding artificial data sets. Japkowicz [35] and Ronaldo [46] have a model for generating imbalanced data sets to control imbalance-related factors, but the output is one-dimensional data sets, which may cause problems when applying learning algorithms. It is discussed in more detail in the following sections. Here we propose an improved data generation method that allows a simple control of data properties.

Our objective is to find out a better ensemble model for classifying imbalanced data sets. The performance is tested by generating various training data with different settings of imbalance-related factors through our data generation model. In more detail, thirty artificial data sets are generated by setting size of data sets, imbalance rate, and data complexity (simple, overlap, small disjuncts). Three re-sampling techniques are chosen: random under-sampling, random over-sampling, SMOTE [13]. They form three Bagging variations.

4.2 Data Sets

Because we try to find out how imbalance-related factors affect final results, we need to generate a number of data sets by setting different parameters to control those factors. In Japkowicz's literature [35], she gave a detailed model of imbalanced data generation. However, generated data is y-axis parallel, so if we use decision tree as our solution, it always has a "better" classification result than other learning algorithms. We do not hope the final performance of algorithm itself is dependent on data set structure, in which way we cannot have a good analysis on different learner methods. Therefore, we present an improved way to generate data set, which is better for comparison among different techniques and analysis of impact of those factors.

4.2.1 Nathalie Japkowicz's Model

Nathalie Japkowicz used her model to create 125 domains with various combinations of data complexity, training set size, and degree of imbalance. Each domain is one-dimensional with inputs in the $[0, 1]$ range associated with one of the two classes. The input range is divided into a number of regular intervals with the same size, each associated with a different class value. Contiguous intervals have opposite class values and the degree of data complexity corresponds to the number of alternating intervals present in the domain. Actual training sets are generated from these backbone models by sampling points at random (using a uniform distribution), from each of the intervals. The number of points sampled from each interval depends on the size of the domain as well as on its degree of imbalance. Figure 4.1 is an example of the backbone model.

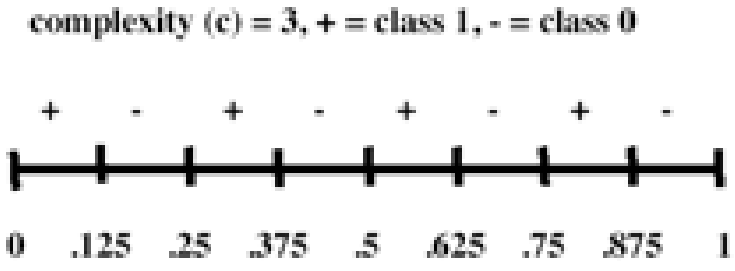

Figure 4.1: Backbone model of complexity 3

Five different complexity levels were considered (1...5) where each level c, corresponds to a backbone model composed of 2^c regular intervals. For example, at complexity level $c = 3$ in figure 4.1, points in intervals $[0, 0.125)$, $(0.25, 0.375)$, $(0.5, 0.625)$, and $(0.75, 0.875)$ are associated with class value 1, while those in intervals $(0.125, 0.25)$, $(0.375, 0.5)$, $(0.625, 0.75)$, and $(0.875, 1]$ are associated with class value -1.

Five training set sizes were considered ($s = 1...5$) where each size s, corresponds to

a training set of size round $((5000/32) \cdot 2^s)$. For example, at the size level of $s = 1$, data set has 312 instances. At the level of $s = 3$, data set has 1250 instances. Actually, we fix those two data set sizes in our experiments.

Five levels of class imbalance were considered ($i = 1...5$) where each level i, corresponds to the situation where each sub-interval of class -1 contains more instances $(32/2^i)$ times more than each sub-interval of class 1.

They used the following equation to generate domains:
$size_x = ((x \times 2^s)/2^c) + (((x \times 2^s)/2^c)/(32/2^i))$

4.2.2 Improved Generation Model

Nathalie Japkowicz's model is y-axis parallel, and having more intervals cannot express data complexity very well. Based on her model, we generate data into three categories: simple, overlap, and small disjuncts. Each domain is two-dimensional with inputs in the $[0, 1]$ range associated with one of the two classes (1 or -1). Data of each dimension is generated in the same way with the "backbone model". The difference between two models is that the second dimensional input in Nathalie Japkowicz's model is actually uniform distribution in any range, while the second dimensional input in improved model is generated according to data set characteristics in the same way with the first dimensional input generation. Here is an example when data complexity $c = 1$.

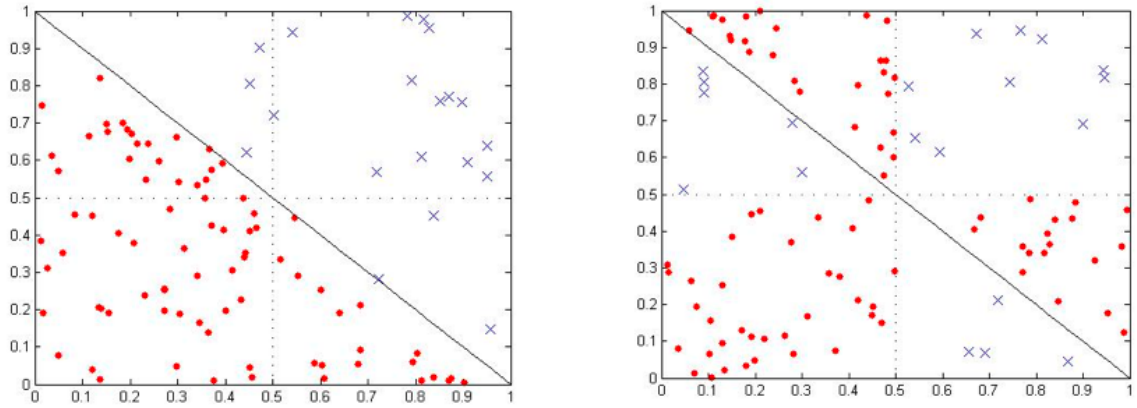

Figure 4.2: Linear separable and small disjuncts(red points: majority; blue crosses: minority)

Each dimension has two intervals $(0, 0.5)$ and $(0.5, 1)$. We assume $(0, 0.5)$ is negative interval and $(0.5, 1)$ is positive. Therefore, all points (x_1, x_2) where both x_1 and x_2 are in interval $(0, 0.5)$ are negative points, and all points (x_1, x_2) where both x_1 and x_2 are in interval $(0.5, 1)$ are positive points. The points where x_1 is in interval $(0, 0.5)$ and x_2 is in interval $(0.5, 1)$, and the points where x_1 is in interval $(0.5, 1)$ and x_2 is in interval $(0, 0.5)$ has three possible conditions after the two areas are divided by line '$y = 1 - x$':

Table 4.1: Summary of data sets in experiments

Data IR	No. of majority class examples	No. of minority class examples
1	294	18
2	277	35
3	250	62
4	208	104
5	156	156
6	1176	73
7	1111	138
8	1000	250
9	833	416
10	625	625

- Linear separable (Left in figure 4.2): points under the line are negative (red), and points above the line are positive (blue).
- Overlap: both negative and positive points exist in those areas.
- Small disjuncts (Right in figure 4.2): points under the line are positive (blue), and points above the line are negative (red).

The proportion of class distribution is the same with the original model when data complexity is 1. Actually, we only consider this complexity in this model, because it already includes three cases. We consider two sizes in our experiments -312 and 1250. For testing data, each class generates 100 points, which means 200 points in each testing data set.

To make our comparison, we generate 30 artificial data sets which have different degrees of imbalance, data complexity and data set sizes. Table 4.1 summarizes the data used in this study. For each data set, it has two continuous attributes and two classes $(1, -1)$, where we define class "1" as minority class. The first five rows are five small data sets, containing around 312 instances. The last five rows are five comparatively large data sets, containing around 1250 instances. Each kind of class distribution has three kinds of data complexity - simple, overlap, and small disjuncts by using our generation model. To test the performance of each model, a test set is also generated for each training set by using same settings. Every test data set has 200 instances totally, and 100 instances for each class.

To test the performance of each model, a test set is also generated for each training set by using same settings. Every test data set has 200 instances totally, and 100 instances for each class.

4.3 Algorithms and Experimental Design

Three "Bagging Variation" models are implemented and compared. First model is implementation of BEV which uses under-sampling technique. However, we find it doesn't work very well in most cases. Especially, it degrades performance of majority data when data set size is not large enough. Then, we proposed to introduce random over-sampling and SMOTE techniques to improve overall performance of "Bagging" as the other two models.

- Under-sampling model (BEV) [39]
  Each part for training individual classifier keeps all minority instances and includes part of majority class instances chosen with no replacement. Number of classifiers is decided by how skewed the data set is. When imbalance rate equals five, it only induces a single classifier without change of class distribution.

- Over-sampling model
  Form each bag by bootstrapping majority data, and over-sampling minority data randomly until it has the same number with majority data within the bag. Number of classifier is set as twenty.

- SMOTEBagging model
  Use SMOTE to generate more minority data firstly, and then apply standard Bagging to new training data set. Two key parameters are involved in SMOTE algorithm number of nearest neighbors k and amount of SMOTE $N\%$. Number of classifier is set as twenty. In our experiments, we choose five nearest neighbors. N ranges between 100 and 500. We present the best one in the final results.

All experiments use decision tree C4.5 as base inductive learner. Pruning scheme used in C4.5 is allowed, because it is not our focus in this paper. Three "Bagging Variation" models are created, which use three re-sampling techniques - under-sampling, over-sampling, and SMOTE. Take majority vote as final result. Comparison is made among each model. We use six outputs to help evaluate model performance: recall value of minority class and majority class, F-value of minority class and majority class, overall accuracy, and G-mean.

4.4 Experimental Studies

One objective of our research is to compare several Bagging variation methods under various imbalance conditions. To make this comparison, we generate 30 conditions by setting different data set properties. To show influence of each factor, we use column charts to organize other factors together.

Comparing different data sets, our experimental results show that overlap data sets always cause the most performance degradation, because overlap area introduce classification difficulty. If ensemble model produces very high recall values of minority class, the recall values of majority class are often relatively low. It is obvious that larger data sets get better values than relatively small data sets, because they provide more useful information. As the data sets are more skewed, other two factors have more influence on final results. Comparing three Bagging variations, BEV model gives the best performance on minority class, but the worst on majority class. For larger data sets, this problem gets less serious, because they could provide sufficient information even after they are separated into several parts with no intersection instead of bootstrap. Over-sampling model and SMOTEBagging model have similar results, and better G-mean values than BEV. They re-sample majority class instances with replacement or by generating new instances.

1. Very Small Minority Class (Imbalance rate = 1)
   Data sets are highly skewed. BEV model produced highest recall values of minority class and lowest recall values of majority class in all six cases. F-value measures the "goodness" of a learning algorithm for classifying a class. BEV didn't always have highest F-measure values. Low accuracy affects BEV's overall performance.

2. Small Minority Class (Imbalance rate = 2)
   Over-sampling model often produced "perfect" recall values of minority class when classifying large overlap data sets. Replicating too many same minority class instances makes model have overfitting problem. SMOTEBagging is a more stable ensemble model from the overall stand of view. Final results are shown in Table 4.

3. Relatively Small Minority Class (Imbalance rate = 3)

4. Mostly Balanced Class (Imbalance rate = 4)

5. Balanced Classes (Imbalance rate = 5)
   For balanced data sets, BEV is a single decision tree model. Over-sampling model is a standard Bagging model having twenty classifiers produced.

We reorganize outputs of each model and each condition into four charts (figure 4.3), showed in the following. Each chart presents one variable in the experiments: Model type, size of data sets, imbalance rate, and data complexity. And the six groups of bars in each chart are the six evaluation metrics, numbered from 1 to 6 in table 4.2.

4.5 Summary

In this work we analyze the behavior of three ensemble models by applying different re-sampling techniques to deal with class imbalance problem. Our results show that

Table 4.2: Evaluation metric list

No.	Metric
1	Recall value of minority
2	Recall value of majority
3	Overall accuracy
4	F-value of minority
5	F-value of majority
6	G-mean value

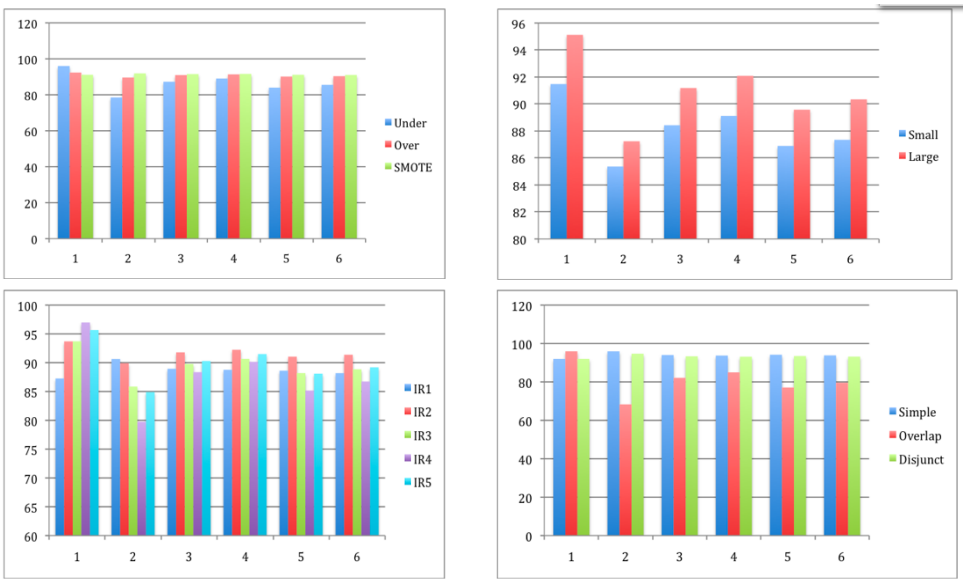

Figure 4.3: (i)Three Bagging variation models. (ii)Small set and large set. (iii)Five imbalance rate. (iv)Three types of data complexity

BEV provides very good recall values in minority, but quite low results in overall performance. Over-sampling model and SMOTEBagging model have better overall performance. These two models have similar results, but SMOTEBagging seems more stable, because it gets new information instead of replicating original instances. Impact of imbalance-related factors is also presented by using an improved data generation model. Overlap always degrades system performance greatly, but not for small disjunct cases. Smaller data set size and more skewed class distribution also influence final results because of insufficient minority data information, especially when data complexity level is high, which conforms to our analysis in previous sections.

Chapter 5

Proposed Work and Experiments

In this section, we propose several ideas and plans for future research. Main experiments are designed in this part. The section follows the inadequacies of previous work and importance of our research closely discussed in the first three sections, but there is not any conclusion in this section.

5.1 Multi-class Imbalanced Data Sets

5.1.1 Multi-class Classification

Most research efforts on imbalanced data sets have traditionally concentrated on two-class problems. However, this is not the only scenario where the class imbalance problem prevails. In the case of multi-class data sets, it is much more difficult to define the majority and minority classes. For example, one class A can be majority with respect to class B, but minority with respect to another class C. Or there are two or more minority classes with respect to one majority class. There are not many works addressing the imbalance multi-class problem. Although multi-class problems can be converted into a series of binary classification problems and then methods effective in two-class learning can be used, researchers usually prefer a more direct solution. Recent studies [58] have shown that some two-class techniques are often not so useful when being applied to multi-class problems directly, especially in the case of imbalance. Some problems are observed experimentally:

- Multi-class cost-sensitive learning is relatively more difficult than that on two-class tasks.
- Higher degree of class imbalance may increase the difficulty of multi-class classification.
- The sampling methods and SMOTE are usually ineffective and often cause negative effect, especially on data sets with a big number of classes.

It is not difficult to understand that multi-class tasks make decision regions harder to be decided. An example can be misclassified in more ways than it might be in two-class tasks. For re-sampling techniques, over-sampling methods, such as SMOTE, make the decision regions larger and less specific by moving borderline towards majority examples; under-sampling methods aim at making decision borderline much clearer by removing "redundant" or "borderline" majority examples. In essence, re-sampling techniques is efficient on imbalanced data sets with binary classes by adjusting classification borderlines. Therefore, complex decision regions in multi-class tasks make those methods ineffective.

5.1.2 Proposed Work

For our research, we are interested in reasons that why re-sampling techniques cause performance degrading for multi-class data and try to find solutions. The advantage of re-sampling is that it will not be limited in algorithms themselves and cost-sensitive information we cannot get. So, it is worthwhile to do more research on multi-class. Considering that there are multiple minority classes, simple under-sampling may not a good choice, because we need to keep useful information as much as possible to help decide classification borderlines. Using over-sampling methods only may also cause problems. The procedure causes overlap easily among multiple classes, thus degrade classification performance. Therefore, I am thinking to use evolutionary prototype selection (EPS) methods to help choose better examples for classification.

Actually, evolutionary algorithms (EAs) [11, 22, 27] have been used for selecting instances especially when data reduction is needed in the data preprocessing phase. The objective is to obtain a subset of training set that allows learners to achieve the maximum classification rate and minimum number of instances in the subset. First, an initial population is created by representing subsets of training data into chromosomes. Second, a proper fitness function is chosen to evaluate individuals. Third, recombine chromosomes through crossover and mutation operators. As a part of fitness function, simple learners need to be constructed for evaluating the subset represented by chromosome of population. 1NN and decision tree (C4.5) are normally used. Figure 5.1 shows how the algorithm acts.

Our intention is to find best data subsets of multi-classes aiming at good performance on both minority and majority examples. Two main steps are involved: generate synthetic examples and evolving subsets of training data. The basic idea of the algorithm is described in the following:

There is a training set $\mathcal{D}$ followed the definition in Section 1.5. There are p possible classes, which means , in which are minority classes and others are majority classes. All training examples in $\mathcal{D}$ could be classified into two sets$\mathcal{D}_{maj}$ and $\mathcal{D}_{min}$, where $\mathcal{D}_{maj}$ consists of all examples which have majority class label, and $\mathcal{D}_{min}$ consists of all examples which have minority class label. Size of data set $\mathcal{D}_{maj}$ is denoted by $|\mathcal{D}_{maj}|$ and $\mathcal{D}_{min}$ is denoted by $|\mathcal{D}_{min}|$ examples, then there is $|\mathcal{D}_{maj}| + |\mathcal{D}_{min}| = |\mathcal{D}|$.

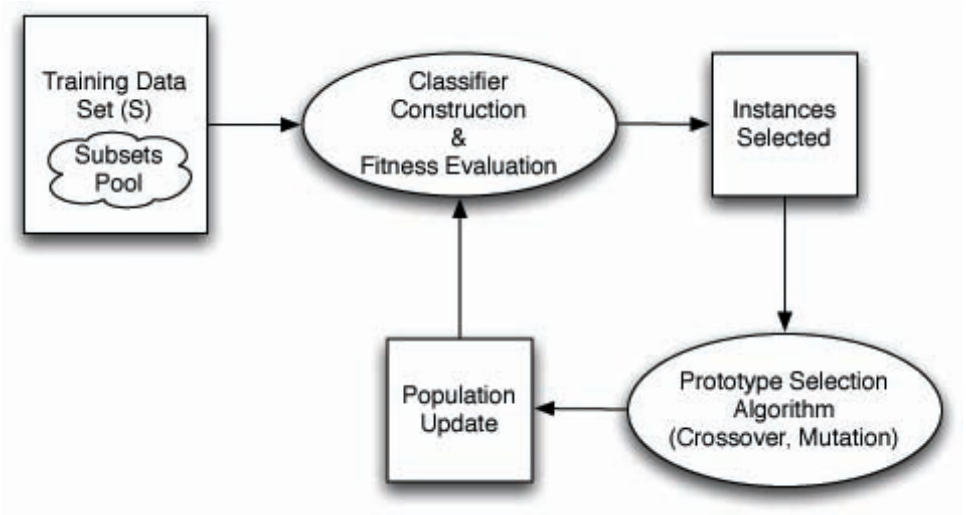

Figure 5.1: Evolutionary prototype selection (EPS) strategy

1. Generate enough examples for each minority class according to the size of majority classes and form data set $\mathcal{D}_{new}$ with $|\mathcal{D}_{new}|$ examples. Now we are trying to find an optimized data subset $\mathcal{D}_{opt}$ from data set $\mathcal{D}_{maj} \bigcup \mathcal{D}_{new}$.

2. Represent the data set $\mathcal{D}_{maj} \bigcup \mathcal{D}_{new}$ by using a binary representation. A chromosome consists of $|\mathcal{D}_{maj}| + |\mathcal{D}_{new}|$ genes with two possible states – 0 and 1. If the gene is 1, its associated example is included in the subset $\mathcal{D}_i$ represented. Otherwise, it is not in the subset. Each chromosome is a possible subset of $\mathcal{D}_{maj} \bigcup \mathcal{D}_{new}$.

3. Let $\mathcal{D}_i$ be a possible subset which is coded by a chromosome. Fitness function is defined as: $Fitness\,(\mathcal{D}_i) = \alpha \cdot G + (1 - \alpha) \cdot per$, where per denotes the percentage of unselected examples in a chromosome, and G is geometric mean - the evaluation criteria denoting accuracy on both majority examples and minority examples.

There are three advantages of this algorithm:

- It is applicable to multi-class data sets including two-class cases. Especially when there are many classes and data complexity is comparatively high, it can remove useless and noisy examples and keep useful ones.

- Class distribution is one part to be evolved instead of setting each class in equal size.

- All information is kept at the beginning instead of under-sampling, and it generates more examples by using SMOTE as part of population to evolve.

5.1.3 Main Experiments

Some important experiments and analysis are included in multi-class analysis:

1. Implement evolutionary algorithm for multi-class imbalanced data sets. Analyze performance on artificial data sets and UCI data sets.
2. Compare performance between proposed algorithm and other re-sampling methods, or other algorithms which could be used for multi-class data sets directly.
3. Analyze advantages and disadvantages of each multi-class technique from theoretical and experimental points.
4. Propose improving or new algorithms based on the analysis to overcome drawbacks.
5. Expand current evaluation metrics to multi-class.

5.2 Negative Correlation Learning on Imbalanced Data Sets

Considering ensemble model is a good way to solve IDS problem, re-sampling methods play an important role in it. Most proposed novel ensemble methods are based on re-sampling training data in different ways. We also start research from algorithm level. In this section, some ideas about negative correlation learning on imbalanced data sets are proposed. How and why NCL works is introduced firstly. Then, an initial idea about how to use NCL on imbalanced data sets is described.

5.2.1 Negative Correlation Learning (NCL)

From previous sections, we know that ensemble models are always used for solving class imbalance problem. And for an ensemble model, its final performance is decided by two factors - classification accuracy of each individual base learner and diversity among ensemble members. It is also clear that error reduction can be achieved by decreasing variance, which is directly proportionate to the diversity or ambiguity in the predictions of the components. Therefore, many algorithms have been proposed to construct a good classifier ensemble by seeking both accuracy of base classifiers and diversity among them. NCL is the most explicit and direct way to increase diversity among ensemble algorithms, and achieves better results. For example, "Managing Diversity in Regression Ensembles" [10] gives a sound theoretical explanation on how diversity works in ensemble and why negative correlation learning (NCL) works. "Evolving a Cooperative Population of Neural Networks by

Minimizing Mutual Information" uses NCL to evolve neural network ensembles and tests on real-world problems [43].

Negative correlation (NC) learning (Liu (1998)) is a neural network ensemble learning technique developed in the Evolutionary Computation literature. Negative correlation learning introduces a correlation penalty term into the error function of each individual network in the ensemble so that all the networks can be trained simultaneously and interactively on the same training data set $\mathcal{D}$. The error function E_i for network i in negative correlation learning is defined by :

$E_i = \frac{1}{N}\sum_{n=1}^{N} E_i(n) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}\left(F_i(n) - d(n)\right)^2 + \frac{1}{N}\sum_{n=1}^{N} \lambda p_i(n)$ and

$p_i(n) = \left(F_i(n) - F(n)\right)\sum_{j\neq i}\left(F_j(n) - F(n)\right)$

where $E_i(n)$ is the value of the error function of network i at presentation of the n-th training pattern. The first term in the right side is the empirical risk function of network i. The second term p_i is a correlation penalty function. The purpose of minimising p_i is to negatively correlate each network's error with errors for the rest of the ensemble.

During the training process, each network i minimizes not only the difference between its output and expected output, but also the diffrence between ensemble output and expected output, so that diversity and accuracy are guaranteed at the same time.

5.2.2 Why Use NCL on Imbalanced Data Set

For imbalanced data sets, negative correlation learning (NCL) could be a good choice for the following several reasons. First, NCL is a successful ensemble technique by inducing diversity among ensemble members explicitly. Hence, it provides us a way to increase diversity by training minority examples. Second, NCL is an incremental learning process, in which way we do not have to mix majority examples and minority examples together during training. Therefore, it is possible to provide us information about what kind of data needed for training, generate corresponding data and continue the training procedure.

5.2.3 Diversity Analysis

Diversity is a main aspect in ensemble learning. As NCL is a neural network ensemble model and some other ensemble models have been used in imbalanced data sets described in section 4, diversity analysis is a part of our research. We try to find out how diversity is related to performance on both minority data and whole data sets, and explore it in order to get better generalization in class imbalance learning.

Before we evaluate how diversity influences generalization error, one problem we have is that there are different diversity measures that can be used in this context.

We will study different measures and give a sound analysis about how diversity can help class imbalance learning problem.

5.2.4 Main Experiments

As a main part of our research, some important experiments and analysis are involved:

1. Check NCL performance on artificial and UCI imbalanced data sets with two classes.
2. Analyze NCL performance by comparing with other ensemble models, such as bagging, boosting, and their variations. Identify advantages and disadvantages. Give improving ideas.
3. Combine NCL with different re-sampling techniques and do the performance comparisons. Identify advantages and disadvantages. And give improving ideas.
4. Check NCL performance on artificial and UCI imbalanced data sets with multi classes. Analyze result from theoretical and experimental points. Compare performances among different multi-class techniques. Compare performances between multi-class and two-class conditions.
5. Diversity analysis on different ensemble models.

Chapter 6

Evaluation and Expected Contributions

This section describes the evaluation of our proposed research questions and the expected contributions to data mining field.

6.1 Evaluation

This thesis focuses on class imbalance learning problem, in which four topics are mainly discussed: ensemble learning on imbalanced data sets, multi-class classification on imbalanced data sets, NCL and diversity analysis on imbalanced data sets, and data heterogeneity problem. For the first topic, some work has been done and is describes in Section 4, but it needs more experiments and comparisons from different aspects give deeper analysis. Other three topics are identified and formulated into uniform expressions. They involve developing novel and improved learning algorithms, and experimental comparisons and theoretical analysis of proposed methods, including prediction accuracy on minority data, prediction accuracy on whole data sets, and diversity analysis. We will test on both artificial data sets and real-world data sets. Main criteria for evaluating whether this research has been successful are:

- Do the experiments produce statistically significant results?
- Do the experimental evaluation criteria achieve expected objectives?
- Do the proposed methods yield positive effects on real-world applications?

6.2 Expected Contributions

The expected contributions of the thesis are concluded in the following:

- Multi-class classification
  One of the main contributions of our research is to show how to use re-sampling techniques to classify imbalanced data sets better which have multiple minority classes. An initial scheme is proposed by combining evolutionary prototype selection method. Other re-sampling or multi-class algorithms will also be included to give a sound and persuasive comparisons and explanation from different aspects.

- New algorithm-level strategies
  Another main expected contribution is to present how ensembles can deal with class imbalance problem. Bagging, boosting, and NCL are included, and combine with different re-sampling methods. We explore ensembles in order to get better recall value on minority data and better performance on both minority and majority classes.

- Diversity Analysis
  As we explore ensemble models, diversity analysis is necessary and helpful in our research. The work is to show how diversity is related to class imbalance learning and explore it in order to help improve current models in class imbalance learning. Diversity of only minority data is considered as one of the main evaluation criteria.

- Sound evaluation strategies
  Multiple evaluation criteria are used to test different aspects of proposed models. Recall values, F-measure, and ROC focus on minority data. G-mean tells us overall accuracy. Especially, we consider execution time in our research as an indicator of proposed approaches during evaluation. A comparison of execution time with current algorithms for imbalanced data sets indicates that which algorithms are better to solve what kinds of applications.

Chapter 7

Research Timetable (Revised)

This section gives a detailed plan for the next two years' research. A short-term timetable is given for the next six months' research. A long-term timetable is for the rest time of study. Tasks in short-term section concentrate on the analysis of ensemble models. All tasks are listed here, but some of them may be removed from the plan depending on the results of prior experiments and discussion.

7.1 Short Term

From September to October 2008

- Check NCL performance on artificial and UCI imbalanced data sets with two classes.
- Analyze NCL performance by comparing with other ensemble models. Identify advantages and disadvantages.

From November to December 2008

- Use Bagging/Booting ensemble model and its variations on real-world imbalanced data sets. Then give an experimental analysis.
- Analyze and compare performance of built ensemble models.
- Determine which diversity measures are suitable to class imbalance learning through experiments.

From January to February 2009

- Diversity analysis on built ensemble models.

- Study multi-class classification algorithms in imbalanced data sets.

March 2009

- Diversity analysis and execution time calculation on built ensemble models.
- Conclude how ensemble models work on class imbalance problem (two classes).
- Write progress report 4.

If we prove NCL or diversity does help classification in IDS, we will do the following:

- Combine NCL with different re-sampling techniques and do the performance comparisons and analysis.
- Propose new algorithms to improve classification performance based on NCL experiments.

7.2 Long Term

From April to June 2009

- Multi-class classification analysis in IDS.
- Check ensemble performance on artificial and UCI imbalanced data sets with multiple classes.

From July to September 2009

- Analyze result from theoretical and experimental points. Compare performances among different multi-class techniques.
- Compare performances between multi-class and two-class conditions.
- Propose improved idea from algorithm-level.

From October to December 2009

- Analyze re-sampling techniques on multi-class imbalanced data sets. Investigate why they may or may not have negative effect compared with two-class cases.
- Implement evolutionary algorithm for multi-class imbalanced data sets. Analyze performance on artificial data sets and UCI data sets.

- Compare performance between proposed algorithm and other re-sampling methods, or other algorithms which could be used on multi-class data sets directly.
- Write progress report 5.

From January to March 2010

- Analyze advantages and disadvantages of each multi-class technique from theoretical and experimental points from data level.
- Propose and experiment improved or new algorithms based on the analysis to overcome drawbacks (data level).
- Write progress report 6.

From April to June 2010

- Combine data-level techniques with algorithm-level algorithms (ensemble models, etc.) to solve multi-class imbalanced data sets.
- Analyze performance on new models including diversity analysis and execution time analysis.
- Prepare for thesis writing.

From July to October 2010

- Finish thesis writing.
- Prepare for progress report 7.

Bibliography

[1] Robert E. Banfield, Lawrence O. Hall, Kevin W. Bowyer, and W.P. Kegelmeyer. A comparison of decision tree ensemble creation techniques. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):173–180, 2007.

[2] R. Barandela, E. Rangel, J.S. Sanchez, and F.J. Ferri. Restricted decontamination for the imbalanced training sample problem. *Lecture notes in computer science*, 2905:424–431, 2003.

[3] R. Barandela, J. S. Sanchez, V. Garcia, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2002.

[4] Ricardo Barandela, Rosa M. Valdovinos, J. Salvador Sanchez, and Francesc J. Ferri. The imbalanced training sample problem: Under or over sampling? *Lecture Notes in Computer Science*, 3138:806–814, 2004.

[5] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. In *Special issue on learning from imbalanced datasets*, volume 6, pages 20–29, 2004.

[6] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–139, 1999.

[7] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[8] Leo Breiman. Bias, variance, and arcing classifiers. *Bias, Variance, and Arcing Classifiers, Technical Report 460, Statistics Department, University of California*, 1996.

[9] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[10] Gavin Brown, Jeremy L. Wyatt, and Peter Tino. Managing diversity in regression ensembles. *The Journal of Machine Learning Research*, 6:1621 – 1650, 2005.

[11] Jose Ramon Cano, Francisco Herrera, and Manuel Lozano. Using evolutionary algorithms as instance selection for data reduction in kdd: An experimental study. *IEEE Transactions on Evolutionary Computation*, 7(6):561–575, 2003.

[12] Nitesh V. Chawla. C4.5 and imbalanced data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, 2003.*, 2003.

[13] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, pages 341–378, 2002.

[14] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6(1):1–6, 2004.

[15] Nitesh V. Chawla, Aleksandar Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *Knowledge Discovery in Databases: PKDD 2003*, volume 2838/2003, pages 107–119, 2003.

[16] Padraig Cunningham. Ensemble techniques. *Technical Report UCD-CSI-2007-5*, 2007.

[17] Thomas G. Dietterich. Ensemble methods in machine learning. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15, London, UK, 2000. Springer-Verlag.

[18] Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, August, 2000 August 2000.

[19] Thomas G. Dietterich. Ensemble learning. *The Handbook of Brain Theory and Neural Networks*, 2002.

[20] Thomas G. Dietterich and Eun Bae Kong. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms.

[21] Pedro Domingos. Metacost: a general method for making classifiers cost-sensitive. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164, 1999.

[22] Dennis J. Drown, Taghi M. Khoshgoftaar, and Ramaswamy Narayanan. Using evolutionary sampling to mine imbalanced data. In *Sixth International Conference on Machine Learning and Applications, 2007. ICMLA 2007.*, pages 363–368, 2007.

[23] Chris Drummond and Robert C. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Proceedings of the International Conference on Machine Learning (ICML 2003) Workshop on Learning from Imbalanced Data Sets*, 2003.

[24] Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 973–978, 2001.

[25] Wei Fan, Salvatore J. Stolfo, Junxin Zhang, and Philip K. Chan. Adacost: Misclassification cost-sensitive boosting. In *Proc. 16th International Conf. on Machine Learning*, pages 97–105, 1999.

[26] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proc. of the 13th. Int. Conf. on Machine Learning*, 1996.

[27] Salvador Garcia, Jose Ramon Cano, Alberto Fernandez, and Francisco Herrera. A proposal of evolutionary prototype selection for class imbalance problems a proposal of evolutionary prototype selection for class imbalance problems. *Lecture Notes in Computer Science*, 4224:1415–1423, 2006.

[28] V. Garcia, J.S. Sanchez, R.A. Mollineda, R. Alejo, and J.M. Sotoca. The class imbalance problem in pattern classification and learning.

[29] Hongyu Guo and Herna L. Viktor. Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *SIGKDD Explor. Newsl.*, 6(1):30–39, 2004.

[30] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing*, pages 878–887, 2005.

[31] Peter E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3):515–516, 1968.

[32] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, August 1998.

[33] Nathalie Japkowicz. Class imbalances: are we focusing on the right issue. In *Workshop on Learning from Imbalanced Data Sets II, 2003*, pages 17–23, 2003.

[34] Nathalie Japkowicz, Catherine Myers, and Mark A. Gluck. A novelty detection approach to classification. In *IJCAI*, pages 518–523, 1995.

[35] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429 – 449, 2002.

[36] Laurikkala Jorma. Improving identification of difficult small classes by balancing class distribution. In *Conference on artificial intelligence in medicine in Europe No8, Cascais , PORTUGAL*, volume 2101, pages 63–66, 2001.

[37] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7, pages 231–238, 1995.

[38] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proc. 14th International Conference on Machine Learning*, pages 179–186, 1997.

[39] Cen Li. Classifying imbalanced data using a bagging ensemble variation. In *ACM-SE 45: Proceedings of the 45th annual southeast regional conference*, pages 203–208, 2007.

[40] Xu-Ying Liu and Zhi-Hua Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Sixth IEEE International Conference on Data Mining (ICDM'06)*, pages 970–974, 2006.

[41] Yong Liu. *Evolutionary Ensembles with Negative Correlation Learning*. PhD thesis, Univ. of New South Wales, Australian Defence Force Academy, 1998.

[42] Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10):1399–1404, 1999.

[43] Yong Liu, Xin Yao, Qiangfu Zhao, and Tetsuya Higuchi. Evolving a cooperative population of neural networks by minimizing mutual information. In *Proceedings of the 2001 Congress on Evolutionary Computation, 2001*, volume 1, pages 384 – 389, 2001.

[44] Larry M. Manevitz and Malik Yousef. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.

[45] Robi Polikar. Ensemble based systems in decision making. In *IEEE CIRCUITS AND SYSTEMS MAGAZINE*, 2006.

[46] Ronaldo C. Prati, Gustavo E.A.P.A. Batista, and Maria Carolina Monard. Class imbalances versus class overlapping: An analysis of a learning system behavior. *Lecture Notes in Computer Science*, 2972:312–321, 2004.

[47] Foster Provost. Machine learning from imbalanced data sets 101. Extended Abstract.

[48] Bhavani Raskutti and AdamKowalczyk. Extreme re-balancing for svms: a case study. *SIGKDD Explorations*, 6(1):60–69, 2004.

[49] David Tax. *One-class classification*. PhD thesis, Delft University of Technology, 2001.

[50] Kai Ming Ting. A comparative study of cost-sensitive boosting algorithms. In *Proc. 17th International Conf. on Machine Learning*, pages 983–990, 2000.

[51] Iban Tomek. Two modifications of cnn. *IEEE Transactions on Systems, Man and Cybernetics*, 6(11):769–772, 1976.

[52] Naonori Ueda and Ryohei Nakano. Generalization error of ensemble estimators. *IEEE International Conference on Neural Networks*, 1:90–95, 1996.

[53] Giorgio Valentini and Francesco Masulli. Ensembles of learning machines. In *Neural Nets WIRN Vietri-02, Series Lecture Notes in Computer Sciences, M. Marinaro and R. Tagliaferri*, 2002.

[54] Sofia Visa and Anca Ralescu. The effect of imbalanced data class distribution on fuzzy classifiers - experimental study. In *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pages 749–754, 2005.

[55] Sofia Visa and Anca Ralescu. Issues in mining imbalanced data sets- a review paper [c]. In *Proceedings of the Sixteen Midwest Artificial Intelligence*, 2005.

[56] Gary M. Weiss. Mining with rarity: a unifying framework. *SIGKDD Explor. Newsl.*, 6(1):7–19, 2004.

[57] Gary M. Weiss and Foster Provost. Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research, 2003*, pages 315–354, 2003.

[58] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. In *IEEE Transactions on Knowledge and Data Engineering*, volume 18, pages 63– 77, 2006.

[59] Xingquan Zhu. Lazy bagging for classifying imbalanced data. In *Seventh IEEE International Conference on Data Mining, 2007. ICDM 2007.*, pages 763–768, 2007.