Loss Pattern Recognition and Profitability Prediction for Insurers through Machine Learning

by

Ziyu Wang

B.S. Applied Mathematics, University of California, Los Angeles, 2014
Submitted to the Center for Computational Engineering
and Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of
Master of Science in Computation for Design and Optimization
and
Master of Science in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2017

Author .... Signature redacted ....
Center for Computational Engineering
May 10, 2017

Certified by .... Signature redacted ....
David Simchi-Levi
Professor of Engineering Systems
Thesis Supervisor

Certified by .... Signature redacted ....
John N. Tsitsiklis
Clarence J Lebel Professor of Electrical Engineering
Thesis Reader

Certified by .... Signature redacted ....
Saurabh Amin
Assistant Professor of Civil and Environmental Engineering
Thesis Reader

Accepted by .... Signature redacted ....
Youssef Marzouk
Associate Professor of Aeronautics and Astronautics
Co-Director, Computation for Design and Optimization

Accepted by .... Signature redacted ....
Leslie A. Kolodziejski
Professor of Electrical Engineering
Chair, Department Committee on Graduate Students

Loss Pattern Recognition and Profitability Prediction for Insurers through Machine Learning

by

Ziyu Wang

Submitted to the Center for Computational Engineering and Department of
Electrical Engineering and Computer Science
on May 10, 2017, in partial fulfillment of the
requirements for the degrees of
Master of Science in Computation for Design and Optimization
and
Master of Science in Electrical Engineering and Computer Science

Abstract

For an insurance company, assessing risk exposure for Property Damage (PD), and Business Interruption (BI) for large commercial clients is difficult because of the heterogeneity of that exposure, within a single client (account), and between different divisions, and regions, where the client is active. Traditional risk assessment models attempt to scale up the single location approach used in personal lines: A large amount of data is collected to profile a sample of the locations and based on this information the risk is then inferred and somewhat subjectively assessed for the whole account. The assumption is that the risk characteristics at the largest locations are representative of all locations, and moreover, that risk is proportional to the size of the location. This approach is both ineffective and inefficient. Thus our first goal is to build a better risk assessment model through machine learning based on clients' data from internal sources. Further, we define a new problem, to predict whether a specific contract would be profitable or unprofitable for the insurance company. This problem turns out to be an imbalance classification, which attracts the second half of our research efforts in this thesis.

In Chapter 2, we first review related literature on state-of-the-art risk assessment models in the field of insurance. Later in the chapter we move to the imbalance classification problems and review some popular and effective solutions researchers have proposed. In Chapter 3, we describe the data structure, provide some preliminary analysis over certain attributes and discuss the preprocessing techniques used for feature construction. In Chapter 4, we propose a new model with the objective to develop a new risk index which represents clients' potential future risk level. We then compare the performance of our new index with the original risk index used by the insurance company and computational results show that our new index successfully captures clients' financial loss pattern, while the original risk score used by the insurance company fails to do so. In Chapter 5, we propose a multi-layer algorithm to predict whether a specific contract would be profitable or unprofitable for the insurance company. Simulation shows that we can accurately label more than 83 percent of the contracts on record and that our proposed algorithm outperforms traditional classifiers such as Support Vector Machines and Random Forests. Later in the chapter, we define a new imbalance classification problem and propose a hybrid method to improve the recall percentage and prediction accuracy of Support Vector Machines. The method incorporates unsupervised learning techniques into the classical Support Vector Machines algorithm and achieves satisfying results. In Chapter 6, we conclude the thesis and provide future research guidance. This thesis builds models and trains algorithms based on real world business data from a global leading insurance and reinsurance company.

Thesis Supervisor: David Simchi-Levi
Title: Professor of Engineering Systems

Thesis Reader: John N. Tsitsiklis
Title: Clarence J Lebel Professor of Electrical Engineering

Thesis Reader: Saurabh Amin
Title: Assistant Professor of Civil and Environmental Engineering

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my thesis advisor, Professor David Simchi-Levi for everything you have done for me. Your guidance, expertise and patience support my journey at MIT and contribute greatly to this thesis. You are my role model and it is my honor to work under your guidance.

I would like to thank my thesis readers, Professor John N. Tsitsiklis and Professor Saurabh Amin, for your patience to me and the precious suggestions to this thesis. You contributed greatly to this thesis and it would not have been done without your revisions and instructions. It is my pleasure to have you as my thesis reader.

I would like to thank my colleague Rui Sun, who cooperated with me in this research project. The project would not have been done as smoothly without your participation. Your talents and insights contributed a lot to the success of this project. You are amazing.

I would like to thank our project collaborators, Venkata Ananth Konda, Rohit Mangal, and Andrew Fano from Accenture PLC, who gave a lot of precious suggestions with their expertise in the field. I would also like to express my appreciation for the Accenture-MIT Alliance that sponsored this research.

I would like to thank our program coordinator, Kate Nelson for always being supportive and my life at MIT would not be as colorful as it has been without your support.

Additionally, many thanks go to my life-long friends Han Wen, Yuhao Zhu, and Yuzhen Yang. Your sincere care and suggestions helped me survive in MIT. You are incredible.

Many thanks go to my roommate and friend Shuai Li. You made my life at MIT an enjoyable and unforgettable one. And I also like to thank all my CDO fellow students who made it a diverse and vibrant community.

I would like to thank everyone in my office for your consistent support, more importantly for the happiness you have brought to me. My life at MIT would not have been as happy as it has been without your companion.

I would like to thank my undergraduate advisors Professor Jianwei Miao, Professor Stanley Osher and Dr. Jerome Gilles who advised me at UCLA and led me to the world of computational science and data analytics. You enlightened me on the value of mathematics and computer science to our society. I benefited a lot from your suggestions about my professional life and career planning.

Finally, I would like to thank my family, especially my father Jiexin Wang and my mother Liwen Zhang. Your love and support are, and will always be, my strongest power to move forward.

Contents

1 Introduction and Background **13**

1.1 Motivation and objective 13

1.2 Thesis Contribution 16

1.3 Thesis Outline 18

2 Literature Review **19**

2.1 Introduction 19

2.2 Risk Assessment Models 19

2.2.1 Existing Risk Systems 20

2.2.2 Conflict between the Limitations and Business Need 22

2.3 Imbalance Classification 23

2.3.1 Problems from imbalance 23

2.3.2 Solutions and Insights 25

2.4 Conclusion 28

3 Data Description and Preprocessing **31**

3.1 Introduction 31

3.2 Data Description 32

3.3 Preliminary Analysis 33

3.3.1 Loss 34

3.3.2 Impact of Industry 38

3.3.3 Original Risk Assessment 41

3.4 Feature Construction 43

3.4.1 Time line 44

3.4.2 Target Variables 45

3.4.3 Input Variables 47

3.4.4 A Full List of Variables 49

4 Client Loss Pattern Recognition 51

4.1 Introduction 51

4.2 A New Risk System 52

4.2.1 The New Risk Index 53

4.2.2 Probabilistic Classifier 54

4.2.3 Numerical Experiments 57

4.3 Conclusion 61

5 Profitability Prediction 63

5.1 Introduction and Motivation 63

5.2 The Multi-layer Algorithm 64

5.2.1 Imbalanced Nature of Dataset 65

5.2.2 Key Methods 66

5.2.3 Algorithm and Computational Results 71

5.2.4 Summary 76

5.3 Another Imbalance Problem 77

5.3.1 Problem 78

5.3.2 Algorithm and Computational Result 79

5.4 Conclusion 82

6 Conclusions and Future Work 85

A Appendix 89

A.1 Algorithm: Adaptive Boosting 89

A.2 Algorithm: SMOTE 90

List of Figures

1-1 Change of Business Volume Over the Past Decade 14

3-1 Distribution of Loss Percentage 37

3-2 Pie Chart of Five Most Occupied Major Industry Groups 39

3-3 Loss Percentage Distribution for Clients Across Industries 40

3-4 Distribution of Risk Quality Rating 42

3-5 Comparison between RQR and Loss Percentage 43

3-6 Time Frame 45

4-1 Flow Chart: Calculation Process for New Risk Index 54

4-2 Distribution of New Risk Index vs. Actual Loss Level 58

4-3 Distribution of Risk Quality Rating vs. Actual Loss Level 59

4-4 Comparison: New Risk Index vs.Risk Quality Rating 60

4-5 Distribution of Risk Quality Rating vs. Actual Loss Level 60

4-6 Comparison: Linear Discriminant Analysis vs. Logistic Regression . . 61

5-1 Profitable Contract vs Non-profitable Contracts 66

5-2 Flow Chart: Adaptive Boosting 68

5-3 Flow Chart for Multi-layer Algorithm 77

5-4 Profitable Accounts vs Non-profitable Accounts (Historically) 79

5-5 Clustering Results 80

5-6 Comparison: Label Distribution in test set 81

A-1 Algorithm: SMOTE 90

List of Tables

3.1 Key information for a typical renewal account 33

3.2 Statistics for Loss 36

3.3 Statistics for Loss Percentage 36

3.4 Major Industry Group Represented by SIC 39

3.5 Five Most Occupied Industry Groups 40

3.6 Statistics for Risk Quality Rating 41

3.7 Conversion Table for Loss Percentage Level 47

3.8 List of Variables in Analysis 50

5.1 Confusion Matrix from Support Vector Machines 72

5.2 Confusion Matrix from Random Forest 72

5.3 Confusion Matrix from first step model 73

5.4 Confusion Matrix after selection 73

5.5 Confusion Matrix for Second Step Model 74

5.6 Confusion Matrix for Second Step after Selection 75

5.7 Confusion Matrix for Third Step 75

5.8 Confusion Matrix for Third Step after Selection 76

5.9 Confusion Matrix for Leftovers 76

5.10 Confusion Matrix for SVM 80

5.11 Confusion Matrix for SVM: First Cluster 81

5.12 Confusion Matrix for SVM: Second Cluster 82

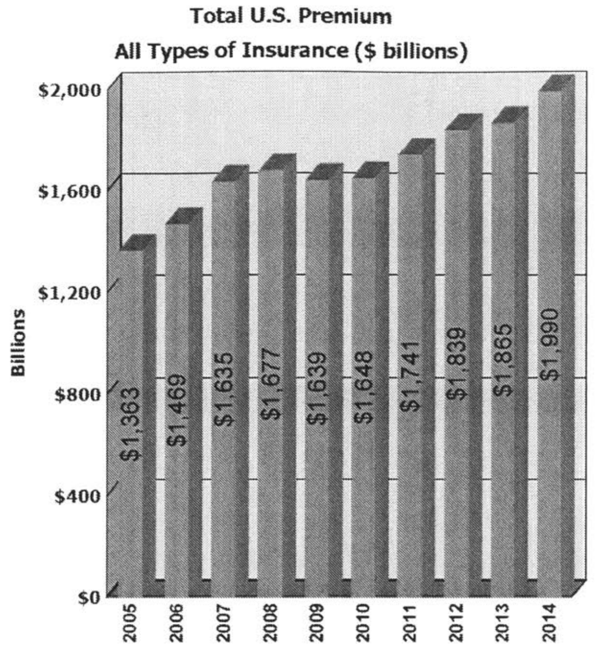
5.13 Confusion Matrix for SVM with Clustering 82

Chapter 1

Introduction and Background

1.1 Motivation and objective

Nowadays the economic environment is totally different from what it was prior to the industrial revolution and the advanced communication age. Enterprises located in every corner on earth are now closely connected and huge business opportunities are hidden under the global market. We all know that benefit comes from the risk undertaken and that people, motivated by the huge interest, are very likely to take more risk than ever in the history. However, the risk sometimes goes far beyond our imagination. In today's world, the consequence of any risky event could be severe, global, and fundamental. In particular, the credit crisis of 2008 gives a wake-up call to the whole world. Baranoff et al. (2009) point out that the 2008 collapse is the result of financially risky behavior of a magnitude never before experienced and its implications dwarf any other disastrous events. We have witnessed the unprecedented worldwide consequences of the crisis and that have hit country after country. Under such circumstance, the importance of risk management has been enhanced historically and one of the most popular means of risk management is to transfer risk to insurance companies. Indeed, we have witnessed a huge increase in business volume for insurance companies in the past decade. Nat (2015) provide the following histogram with exact values of total premium charged, which explicitly shows that the business volume of the US insurance market has increased for nearly 50 percents over



(Premiums from Property, Life, Fraternal, Health and Title Annual Statements plus State Funds for Property and Health)
Source: National Association of Insurance Commissioners

Figure 1-1: Change of Business Volume Over the Past Decade

the last ten years.

Intuitively, the insurance industry will have a substantial increase in the volume of business after a worldwide risk event. Smith et al. (2007) demonstrate the intuition by presenting the fact that there was a tremendous primary rate increase for insurance market from 2001 to 2005 due to losses from the 9/11 terrorist attack as well as deteriorating underwriting results from the soft market of 1997 to 2001. In fact, the phenomena reflects the essence of insurance. As pointed out by Best (2008), insurers make money by managing various types of risk for individuals, municipalities and corporate entities. However, where there is risk, there is uncertainty,and where there is uncertainty, there is exposure to volatility. Thus it is important for insurance companies to quantify the risk exposure for each of their potential business. In other words, they need an accurate risk assessment that helps underwriters balance the insurance company's profitability with clients' potential need to use the policy. If they overestimate their clients' risk and charge too much for premium, they will loss clients; on the other hand, if they underestimate clients' risk, they will be exposed to huge risk which might cause claims with amount much higher than the premium charged. Especially for business related to corporate entities, either situation would lead to a huge loss in revenue and thus an accurate risk assessment is the core part of the business.

Traditionally, Business to Business (B2B) insurance companies would hire a third-party engineering company to help with their risk modeling. Through on-site investigations, engineers would provide professional engineering reports based on their trips to the clients' factories, offices, warehouses and other properties. Finally a risk score would be calculated for each customer and underwriters would use the score to calculate premium rates for policyholders. The methodology worked quiet well a few decades ago when insurance companies limited their business to only a few industries in their local areas. Unfortunately, economic globalization and industrial diversification bring huge challenges to this risk assessment model. Currently, our project collaborator provides insurance products to commercial firms across hundreds of industries through out the world and it is impossible for the company to afford on-site engineering investigations. Based on their experience in personal lines, analysts from our collaborator attempt to scale-up the single location approach to assess risk exposure for commercial clients. They assume that risk characteristics at the largest locations are representative of all locations and that risk is proportional to the size of the location. Engineers perform on-site investigations for a sample of the largest locations and analysts then scale-up these investigations and assess risk for the whole account. The assessment is somewhat subjective and insurers believe that the method is both ineffective and inefficient.

A costly but ineffective risk assessing model is definitely not acceptable for insurance companies and thus insurers are in urgent need of a more efficient and accurate assessment of their potential risk exposure. However, the challenge here is that we are strictly limited by the size of reliable data source. Among all data sources, only the insurer's internal database is reliable and other data such as the client's submission data are marked as unreliable. Thus the main concern is whether we can provide an accurate and efficient method to predict the insurer's potential financial risk exposure based only on their internal data sources. Further, our collaborators define a more practical task that based on the same data, we directly predict whether a specific contract would be profitable or unprofitable for them.

In this thesis, we first work on the two challenges mentioned above and we focus only on renewal accounts with record in the internal database. Specifically, we focus on the companies that have already cooperated with the insurance company for at least a year so that we can conduct analysis base solely on the internal dataset. We propose two models and first of which would show the financial loss pattern and the second would be used for the gain-loss labeling. While developing the second model, we also research on the class-imbalanced classification problems. Modeling and numerical experiments are all implemented in the statistical computing environment R of version 3.1.1 GUI 1.65 Snow Leopard build. We run our algorithms on a MacBook Pro with 2.5 GHz Intel Core i5 CPU and system OS X 10.8.5.

1.2 Thesis Contribution

The contributions of our work are two-fold. The first one focuses on insurance risk assessment and the second lies in the field of machine learning classification algorithm. Research on risk assessment for insurance companies mostly lies in credit risk in personal lines and there has not been much attention on business property insurance. Generally, previous research in business level focus on a specific type of risk exposure and the models are developed based on external data from public resources. Specifically, the majority of related research focuses on assessing risk exposure for natural disasters. Hsu et al. (2011) provide a probabilistic model based on historical events to assess flood risk in Taiwan and their model is developed based on local data obtained from the Taiwan government. Tsai and Chen (2011) present a mechanism for typhoon- and flood-risk assessment for the local hotel industry. To the best of our knowledge, in the field of insurance, there has not been any research on risk assessment models that cover all types of risks for clients across all industries. We develop a new risk assessing system to address the issue, with the explicit goal of capturing the actual financial loss pattern across all risk types for clients in all industries. Specifically, we define a new risk index to represent the financial risk level for all types of risks and develop a model to calculate the index based solely on business-related data from internal database. The new risk system provide a reliable reference to underwriters in setting up of insurance policies and calculating the premium rates.

Secondly, we build another model for the profitability prediction, which we approach as a binary classification problem. In particular, we are dealing with an class imbalanced dataset and it is usually very difficult for classical learning algorithms to detect events in the minority group. There has been diverse research on the class imbalanced dataset to help detect events in the minority group.Chawla (2005) discuss popular sampling techniques used for balancing the datasets and Longadge and Dongre (2013) provide a comprehensive review for the techniques developed for the class imbalanced datasets. In particular, Chawla et al. (2002) proposed the famous SMOTE algorithm, which incorporates oversampling and undersampling techniques to re-balance the dataset. Later Chen et al. (2004) propose to use cost sensitive learning and sampling technique to improve the performance of random forest. Tang et al. (2009) propose to incorporate different re-balance heuristics including cost-sensitive learning, oversampling, and undersampling in Support Vector Machines (SVM).

Based on our observation of the dataset and characteristics of the methods mentioned above, we propose a multi-layer classification algorithm which incorporates the SMOTE method and adaptive boosting techniques. We describe in detail how the multi-layer algorithm works and show the effect by simulation results. We show that our algorithm can improve both the recall and precision of minority group by more than 75 percent for more than 85 percent of the data points. We extend the research to imbalance classification by defining another imbalanced problem. We observe that two of the independent variables are much more important than others in determining the class labels and thus propose a simple method to help classical Support Vector Machine to accurately detect the events from minority group. Particularly, we would show the improvement of recall percentage for more than 50 percent in the minority group with satisfying precision by incorporating unsupervised learning technique into the classical Support Vector Machines (SVM) algorithm. Our method is simple, easy to operate; more importantly, it strengthens our confidence in applying hybrid modeling to help with machine learning tasks.

1.3 Thesis Outline

This thesis consists of six chapters. The remaining parts are organized as follows. In Chapter 2 we review existing literature both in insurance risk assessment and in binary classification for class imbalanced datasets. In Chapter 3 we describe the data structure, show some preliminary results from statistical analysis, and discuss the data preprocessing techniques involved in our analysis. In Chapter 4, we present our work on the development of a new risk assessment system. We define the index, show the modeling process and associated theoretical basis and compare the index performance with the original risk index by numerical experiments. In Chapter 5, we present the second part of our research, the model to label clients as profitable or unprofitable. We present details that show how our proposed algorithm outperforms the traditional classifiers such as Random Forests. Later in this section, we will show another algorithm that we develop for the second classification problem. Our method is designed to improve traditional Support Vector Machines, as applied to our case; computational results clearly show the effectiveness of our proposed method. Finally, we conclude and provide possible directions for future research in Chapter 6.

Chapter 2

Literature Review

2.1 Introduction

As stated in the previous chapter, there are two major concerns in this thesis; to build an effective risk assessment system and the imbalance classification problems initiated by the prediction over profitability of clients. The contents in this chapter are arranged accordingly. In Section 2.2 we first review related literature on state-of-the-art risk assessment models in the field of insurance. In particular, we want to show the way scientists approach the problem of risk assessment for insurance companies and the performance of their models when applied to real world cases. Later in this chapter we move to our second concern of imbalance classification. In Section 2.3 we will show some popular and effective methods scientists used to accurately label elements in the minority class. Especially we want to show how the improvements are made and the sacrifice behind the gains.

2.2 Risk Assessment Models

In this section, we show the most recent works on risk assessment in the field of insurance. We will show the effectiveness of existing models in some restricted application areas but more importantly we want to show the limited capacity of these models when applied to real world cases. One important observation to note is that all existing models are approaching insurance risk assessment from the same direction and thus limited in the same way. Thus we point out the necessity of a new risk assessment system to solve the problem in a new perspective and to break the limitations on the traditional risk assessment models. Due to the limited number of existing literatures in the field, we are only able to find a few relevant papers on it but still we believe that these limited amount of papers would help us to understand the traditional risk assessment models in insurance industry.

2.2.1 Existing Risk Systems

There have been some credit risk assessment models for life insurance or car insurance and indeed most of the literatures focus only on the personal insurance industry and much less efforts are found for Business to Business (B2B) insurance. Thus it is hard for us to make an inclusive and comprehensive review for the area since there might be some unpublished works or private works which are not yet available. However, we still feel that the following reviewed papers are quiet representative and they can at least reflect the traditional approach for relevant works. One important observation worth noting is that we have not found any work or existing risk assessment system that can represent all risk types, which is referred to all risk in insurance.

Traditional approaches for risk assessment in insurance usually focus on a specific type of risk, such as earthquake or flood and most work would use public available data for predictions. The observation is in accordance with what we have mention in the previous chapter, that insurance company assessing their risk exposure based on external engineering data. Examples of relevant work would be Hsu et al. (2011), who develop a probabilistic model to predict the potential flood risk in Taiwan. Prior to the work of Hsu et al. (2011), Apel et al. (2004) also develop a model to assess flood risk for insurance use and Spence et al. (2008) with their earthquake estimation system applied to insurance industry. These works are all built upon external datasets such as the public available weather data, government owned flood frequency data, and geographical and engineering related data. These works are very impressive in their predicting ability and the accuracy.

Researches involved in the above work usually build several sub-models for their predicting system; an engineering model related to the building types or construction areas would be used for vulnerability analysis and a natural event model would be used for predicting the occurrence of a specific natural disaster. We feel that the work of Hsu et al. (2011) could be quiet representative. They build their flood risk engineering model based on the insurance profile data which include the building type, location, height and other relevant information. According to the experience provided by experts from the field, these data are all submission data or data extracted from on-site investigations. Either data source would cause a potential problem to the final risk predicting system. Submission data are provided by clients and insurers would usually label these data as low quality since they are not precise and sometimes even fake; on the other hand, if insurers really want to build their models upon these data, they would have to hire engineers for on-site investigations and as we have mentioned previously, this process would be extremely costly and inefficient. Another important part of the traditional predicting system is the model for natural event and in the case of Hsu et al. (2011), they develop two models for rainfall events and hydraulic events respectively. These models are well-developed and the data come from highly authoritative agencies. However, we have to note that these models can only be applied to the specific type of risk; in other words, the rainfall and hydraulic models can never be used to predict events such as earthquake or tsunami. Thus the predicting system is strictly restricted to a very narrow and specific risk type and the capability of the system is thus affected greatly.

Despite of the above flaws, we find a very attractive advantage in the work of Hsu et al. (2011). The last part of their model is the financial loss model and the predicted flood events data are finally used for estimating the direct and indirect financial losses caused by the events. Their work is directly related to insurance concepts such as the insured loss. This big advantage attracts our attention and it immediately differentiate the work with all other works. Indeed financial loss estimation is the most important and sometimes the only concern of our collaborator. This insight helps us in defining our problem and we will show in the next several chapters that the final output of our risk assessing models are all financial loss oriented.

2.2.2 Conflict between the Limitations and Business Need

In Section 2.2.1 we briefly mention the limitations and potential problems of the traditional risk assessing models developed for insurance companies. We want to first summarize the problems and limitations here and later point out the conflicts between the limitation and some current business need. Two major limitations are mentioned above, first one caused by data quality and second one caused by the application scope. Data quality could potentially cause huge problem in the prediction system; without enough accurate data, it will be hard for systems to capture the loss pattern. For insurance companies, it is impossible to build a large database for high quality engineering and geographical data. Thus a more reliable system should be built upon a more accessible database and the most reliable data source is the internal database managed by the insurance company itself. Later we will show how we use the internal data from our collaborator to assess risk for their clients. The limitation on the application scope is caused by the problem definition itself; the problem is originally defined to assess risk associated with a specific natural disaster and thus we should not expect the model to do more. But assessing the risk exposure to flood or earthquake is not enough for insurers to make decisions and a more comprehensive model is always better for them.

As we mention in the previous chapter, currently insurance companies are expanding their business across all industries around the world. This is the result of globalization and the tendency is irreversible. For insurance industry, it means that insurers are exposed to a wide variety of risk, not restricted to natural disasters but also local security conditions and the economical environments. Thus if we still follow the traditional way of risk assessment, we would need hundreds of models built upon a vast dataset with high quality. This could be imagined ideally but never happen in reality. Thus there is an urgent need for a new risk assessment method, with the ability to cover all types of risks and at the same time built upon currently available and affordable datasets. This is the direct motivation of our project and we will show

how we fulfill the business needs in the next several chapters.

2.3 Imbalance Classification

In this section, we would focus on the class imbalance classification problems; in particular, we want to show some effective methods and models designed for the problem. We will first talk about the imbalance classification itself and point out the common issues associated with the problems. We will then show some state-of-the-art solutions against the imbalance problems and provide some insights into the methods. Later in Chapter 5 we will show how we take the advantages of a method shown here the discussion of profitability prediction.

2.3.1 Problems from imbalance

In this part, we will discuss about the imbalance classification problems. In particular, we will show the problem formulation, concerns over the problems and some important issues associated. Imbalance datasets are very common in our daily life and usually the imbalance ratio is quiet large. In a sunny autumn morning in Boston, you will find much more trees with yellow or red leaves than trees with green leaves; in oceanography and meteorology, the occurrence frequency of tsunami is very small; in our society, there are very few billionaires and the majority are middle class. When we apply machine learning techniques to these different fields, we will inevitably work with these imbalanced datasets. A common task in machine learning is classification and we call the labeling over these imbalanced datasets as imbalance classification problems. A example of such problem would be to identify or predict which trees would stay green during the autumn in Boston.

Intuitively imbalance classification would be harder than the usual more balanced problems and Ali et al. (2015) observe through experiments that relatively balanced distribution between classes in datasets would generally result in better performance of decision trees. However, essentially Imbalance classification is just the same as all other problems except that target variable is dominated by one of the class labels and thus the other class would be insufficiently represented. The under-representation is not a problem itself; nevertheless, as pointed out by Ali et al. (2015), most of traditional classifiers are accuracy driven and the focus lies only on the overall accuracy. Consequently, we would expect traditional algorithms to label everything as the dominant class in order to reduce the overall estimation error. There are also some other minor problems associated with the imbalance problem, but the major hindrance is caused by the imbalance nature itself. Although people are still not sure about the degree of imbalance class distribution to which it would hinder the performance of traditional classier, our observations and theoretical inference leads directly to the conclusion that it is generally hard for common classifiers to deal with minority class members. Ironically, we are indeed more concerned about minority class and usually the need is to identify those elements. If we are dealing with a dataset whose ratio between the majority and minority class is 10 to 1, then even if a classifier predict all data points to be in the majority class, the overall accuracy would be 90 percent. But in the presence of imbalance, overall accuracy is not able to tell much in the minority class. Accuracy as high as 90 percent does not solve anything meaningfully. In the above example of tree labeling, what we really want to know is which trees would stay green and if the algorithm label every tree as yellow or red, the project would then become meaningless. This is the major motivation of the solutions for imbalance classification problems and most of them are design in the way of sacrificing metrics for majority class in order to improve the precision and recall for minority class.

Imbalance classification can be binary or multi-class tasks. In this thesis, we would only focus on the binary imbalance classification problems. Multi-class classifications are essentially more difficult than binary cases and adding the imbalance to the problems would make them even harder to solve. However, methods over the binary problems can be extended to the multi-class cases and ideas behind the innovations should be the same. For detail discussion on multi-class imbalance problems, please refer to Seliya et al. (2008) and Sahare and Gupta (2012). Wang and Yao (2012) also provide some insightful discussions over the issue.

2.3.2 Solutions and Insights

In this section, we present some of the most popular and effective methods people have been using to tackle the imbalance classification problems. We will briefly show techniques from different categories, their basic principles, and the effect they would have to the imbalance problems. Later we will give the major take-aways from these techniques and provide some insights about the methods. Based on these findings and observations, we will use the techniques indirectly to fulfill our needs in order to improve the performance on our specific imbalance classification problems. Here we only give a brief overview of the methods and for the two methods we include in our own algorithm, we will discuss them in detail later in Chapter 5.

State-of-the-art Solutions

Most of the techniques being used against the imbalance classification problems belong to either of the two major categories, the data level approaches and the algorithm level approaches. Ali et al. (2015) and Longadge and Dongre (2013) provide comprehensive reviews on methods in these two categories. Data level approaches are applied in data preprocessing and popular methods include sampling and feature selection. Algorithm level approaches are taken during the actual labeling process and examples of methods include cost-sensitive learning algorithms and ensemble algorithms.

The majority of data level approaches aim to re-balance the dataset and thus alleviate the effect of imbalance on the minority class and these methods are referred as the sampling methods. There are two types of sampling techniques, oversampling and under-sampling. Oversampling duplicates examples in the minority class and thus increase their exposure to the classifiers while under-sampling essentially removes a portion of examples from the majority class to reduce its exposure. They both achieve the effect of re-balancing and reduce the imbalance in original dataset. Sometimes people use these two methods together for better rebalanced effect.

Perhaps the most famous oversampling technique is the Synthetic Minority Oversampling Technique proposed by Chawla et al. (2002), which is often referred as

SMOTE. It is called synthetic since it adds synthetic minority data points to the training set based on the original minority members. SMOTE is an important technique we use to solve our problem and please refer to Section 5.2.2 for detailed description of the method. Although over-sampling methods can increase the exposure of minority class, the method does not provide any new information to the learning algorithms and thus might sacrifice precision for the improvement of recall.

Most of the under-sampling techniques are comparatively simpler in structure and much less inexpensive as for computation cost. Unlike oversampling which needs to add similar data to the feature space, under-sampling undertakes a much simpler task; all they need to do is just to remove some data points from the majority class and usually the removal is determined at random. Thus the potential risk of under-sampling is that by randomly removing data points, it might remove important information contained in the original dataset.

Compared to sampling techniques, feature selection is less popular and more complicated. More importantly the method is case-sensitive and thus it has gained much less exposure to us with only a little literature on this subject. The basic idea behind the method is to measure the relevance of feature and suggest highly-influential features which contain intrinsic information and discriminant property for classification. At the same time, the method could also help to remove redundant or noisy data and thus improve the algorithms computationally. Yin et al. (2013) explore the use of feature selection in high-dimensional imbalanced datasets; however, as they point out, the application of feature selection in imbalanced problems is under-explored. Thus we do not see many usages of feature selection in the field.

Other methods approach the imbalance problems from an algorithmic perspective. Some of them aim to revise existing classifiers such as Support Vector Machines to fit special requirements for specific problem settings; an example is the z-SVM proposed by Imam et al. (2006). Another part of algorithms in the category aim to improve performance by working on the cost functions. These methods are referred to as cost-sensitive methods and the basic principle is to penalize the classifiers when they mis-classify data points from the minority class and thus drag more of their attentions to the under-represented group. Cost-sensitive methods are developed upon the existing algorithms and algorithms such as the cost sensitive Support Vector Machine has been developed. The last part lying in the algorithm approaches are perhaps the methods that we are most familiar with, called the ensemble methods. Instead of using a single classifier to label data points, these methods train several different classifiers and then integrates their results to determine the final result. More common name for the methods are boosting and bagging. There are some famous ensemble learning algorithms and adaptive boosting, bagging and random forest are among them. We include a variant version of adaptive boosting in our algorithm and detailed description of the method and the way we use it would be discussed in Chapter 5.

In general, data level methods tackle the imbalance problem directly by re-balancing datasets while algorithmic methods are usually designed based on the existing algorithms. Both methods could be useful to solve problems raised by imbalance but different methods would fit into different scenarios and our choices should be made upon the actual problems. Ali et al. (2015) provide a list of methods grouped by categories and we believe the list is quiet comprehensive.

Insights

In the previous section, we briefly introduced some of the most popular methods against the imbalance classification problems. Although they have been applied to many different problems and successfully tackle the hindrances, these are all general methods. The extent to which they could help with our problem could be large or small. Thus as some of the researchers have proposed, we consider using a hybrid of the methods to solve our problem. There is no free lunch and whatever method we choose to use, we will need to pay the price. We notice that oversampling would increase the exposure of minority class and thus increase the recall percentage but it would hurt the precision in labeling of majority class; boosting methods emphasizes on data points being mis-classified and thus might not perform well as measured by recall percentages. If we can find a way to hybrid some of the methods together and make efficient use of their advantages, we might be able to get better results.

Also recently there have been some other methods, which proposed to incorporate clustering techniques to help with classification tasks. We believe these innovative thoughts are very insightful and they could potentially help greatly with the imbalance classification problems. Indeed one of our proposed algorithm is based on these ideas. Our method is inspired by Lin et al. (2014) and we will introduce it later in Section 5.3. We will see that the improvement over traditional Support Vector Machine is indeed very surprising, with more than 40 percent improvement of recall percentages over the minority class.

2.4 Conclusion

In this chapter, we reviewed the traditional approaches for insurance risk assessment. We discussed the effectiveness of these methods and at the same time pointed out their limitations. In particular we showed the conflict between the lack of ability of the current insurance risk assessment models and the greedy business need of the field. Traditional risk assessments are well designed and they could be very accurate provided enough investments and time; however, those models could only be applied towards a special risk type, such as earthquake, flood or some other natural disasters. But the current business need is to effectively and efficiently capture all risk, not restricted to natural disasters. Thus we pointed out the motivation of our research project and introduce the main focus of our risk assessing model, to capture all risk across all industries around the world by using only business-related data. Later in the chapter, we jumped to the field of machine learning and focus on the imbalance classification problem. We first introduced the problem settings and showed the potential issues associated with it. We then briefly introduced the existing state-of-the-art techniques against these imbalance problems. We discussed two major categories of solutions, the data level approaches such as oversampling and the algorithm level approaches such as the cost-sensitive learning techniques. We illustrated the way they work, the basic principle supporting the methods and potential issue brought. Based on these introductions and discussion, we briefly went through our thoughts over the techniques and the key takeaways; in particular, we presented two possible ways of applying these techniques to our own problems.

Chapter 3

Data Description and Preprocessing

3.1 Introduction

In this chapter, we describe the structure of our data, point out our key observations and provide some preliminary analysis. Based on the analysis, we would then discuss the data preprocessing techniques used to construct features that would later be used as input variables for the risk assessment modeling and gain-loss classification.

The data of this project comes from a giant global insurance and reinsurance company. It operates in over 60 countries, providing products to commercial firms across various industries throughout the world. With its huge business volume, the company records and generates a large amount of data over the past decade.

We have been able to access a part of the company's database which contains data for a portion of its clients from 2008 to 2013. Our research started at end of 2014 and we first develop our models based on the first part of the data, as it was the only available dataset at the time. Later when we achieve satisfying result on the data, the insurance company transfered the second part of the data to us. We then validated our models on this part of data and achieved consistent results. In this chapter, we will only give description for the latest dataset and the previous dataset is similar to it. Again, as we mentioned in the previous chapters, all these data are from renewal accounts and our research does not apply to new clients who have never cooperated with the insurance company before.

This Chapter is structure as four sections and each of which would discuss a different aspect. We would present a detailed description for the data in Section 4.2 and then give more insights about the data by showing results from preliminary statistical analysis in the following Section 4.3. Finally we would discuss the preprocessing techniques involved in this research and present how we construct features with these techniques in Section 4.4. A full list of features will be provided at the end of this section.

3.2 Data Description

When a new client comes to the insurance company, it will be requested to fill out required forms and submit information related to all aspects of itself. This part of data is called the submission data and it will contain information such as the location, total insured value, and line of business. All these information will be recorded in a separate file as a part of the database. According to the standard procedure, the insurance company will then hire a third-party engineering company to perform on-site investigation for a sample of locations of their clients and engineers will then generate engineering reports for each of the clients. The reports will then be recorded as another part of the database. The underwriters will then generate contracts for clients and the insurance company would decide whether to transfer a portion of the risk to another insurance company by signing a reinsurance contract. The contract data such as the premium charged and reinsurance premium paid are also important parts of the database. The formal cooperation is established since the effective date of the contract. As the business goes on, losses may occur and thus claims would then be filed. Every claim would be recorded and put into the database.

A renewal account would have all the files mentioned above related with it. In table 3.1 we list the essential information for a typical renewal account recorded by the insurance company. Explanations for each of the attributes are also provided in the table.

Notice that we do not list any information related to engineering report in the

Table 3.1: Key information for a typical renewal account

Name	Description
Reference Number	The internal index assigned to the client
Region	Location of the company
Total Insured Value (TIV)	The total amount insured for the client
Property Damage (PD)	damage to property caused either by people other than its owner or by natural disaster
Business Interruption (BI)	the loss of income that a business suffers after a disaster
Premium	The amount of money charged to the client
claim amount	The amount of money paid by the insurance company to the client for the claim filed
Standard Industrial Code (SIC)	A four digit code assigned by the U.S. government to the client, specifying the primary business of the client
Share percentage	The percentage that the insurance company is responsible for loss incurred by the client
Risk Quality Rating (RQR)	Risk score from 0 to 100, showing the risk level of the client; also the original risk index used by the company
Catastrophe loss	Amount of money paid to the client because of natural disasters such as earthquake and floods
Reinsurance premium	Amount of money paid by the insurance company to another reinsurance company for transferring risk
Reinsurance claim amount	Amount of money paid by the reinsurance company to the insurance company for each claim

table above. We are aiming to break the traditional risk assessment model and thus we conduct our research solely based on business-related data. It is also important to point out that we do not use submission data for analysis as well. Clients are supposed to submit their information honestly but there could potentially be some customers who is not willing to let the insurance company know the true situation or they may want to hire a certain part of the information in order to get lower premium rates. Although these are rarely happened, we believe it is still a hidden risk. An accurate and reliable risk assessment system should consider all hidden risk underneath. In the database, only the contract information and the claim information are pure internal data and internal data are much more reliable than data recorded from outside source. Thus we develop our models based only base on internal dataset and this also explains why we focus only on renewal accounts.

In this thesis, there might be terms or attributes with names that are different from what is used in the industry. Although we are trying to solve the issue and have already confirmed the attributes name and explanations with insurance experts, we do not have expertise from the field. However, we will always provide definitions or explanations for the attributes we used throughout the thesis.

3.3 Preliminary Analysis

In this section, we will present some preliminary results from statistical analysis on certain aspects of the dataset. We first give a summary of the data involved in our research and then in the following subsections we will proceed to provide more detailed statistical analysis.

Here are some general statistical information about our dataset.

- We have in total of 13796 contracts over six years, from 2008 to 2013
- The contracts belong to 3035 different clients and each of them has in average 4.55 contracts
- These clients are spread in 35 different countries and transactions involve 21 different currencies
- At the same time, we are provided with 15660 filed claims related to the contracts
- The claims are filed by 1791 clients and each of them has filed in average 8.74 claims
- Among all 15660 claims, 3610 of them are not paid, meaning these claims are under deductible
- For claims with payment, catastrophe count in average 13.67 percents of the amount

3.3.1 Loss

Among all attributes in the dataset, loss is the one that insurance companies concern the most. If a client does not incur any loss during their contract year, the insurance companies would achieve high profit margin for the contract; however, if a client suffers huge loss during the contract year, the insurance companies might have to pay more than the premium charged and thus loss money for this contract. The importance of loss for insurance companies initiates us to build a risk index directly associated with financial loss. Unfortunately, it is impossible for us to calculate the exact loss happened to the clients in each year based on the dataset provided.

In order to explain the obstacles that hinder us to come up with an accurate loss calculation, we need to first briefly describe the operation models in insurance. When underwriters sign contract with their clients, they will explicitly state the amount of total insured value (TIV), the portion that their insurance company is responsible for (share), and the amount of money that clients must pay before insurer will pay any expenses (deductible). Usually the total insured value (TIV) will be divided into several levels (layers) and one insurance company will only undertake a certain percentage in each level (layer). When an event happens, clients will report to the insurance companies and these claims will be processed based on the above information addressed in the contract.

Internal data contain the total amount of money paid to the clients for each filed claim and the number of layers for each contract. However, the data do not show the exact component and we do not know how much is paid for each layer. More importantly, in some cases when the loss is under deductible the insurance company would pay nothing. Nevertheless, in these cases loss events do happen but we do not know how much the loss is.

Although we are not able to calculate the exact amount of loss for clients, we still can estimate the amount of loss based on the information provided. Loss is a relative concept, and we propose here two ways of estimating the losses depending on our objectives. As we have stated in the previous sections, we are first aiming to capture the clients' loss pattern and thus we need to calculate the actual loss incurred by a specific company in a specific year. In this case, the estimated loss will be calculated by the following formula $\frac{\text{Total amount paid by the insurance company}}{\text{Average share percentage in all layers}}$. For example, if the insurance company pays one of its clients 100 dollars in year 2011, the total insured value is divided into three layers, and the shares for each layer are 0.04, 0.05, 0.06. We first calculate the average share percentage, $\frac{0.04+0.05+0.06}{3} = 0.05$. Then we can get the estimated loss for the client in 2011 as $\frac{100}{0.05} = 2000$. Our second objective is to predict whether a specific company will be profitable or unprofitable for the insurance company and thus we are concerning the losses for the insurance company. We need to calculate the potential amount paid by the insurance company. In this case, we need to involve reinsurance into our calculation. We first calculate the net amount paid by the insurance company, which will be the amount they pay to their client subtracted by the amount they received from the other reinsurance company. Then the loss for this case will be calculated by $\frac{\text{Net amount paid}}{\text{Average share percentage in all layers}}$. We use the same example and we know that the other reinsurance company pays the insurance company 20 dollars. Thus the loss brought to the insurance company by the client is $\frac{100-20}{0.05} = 1600$. Use the appropriate estimation of losses is extremely important because it is directly related to our objectives.

As we can see, the second method provides a much more accurate estimation compared with the first method. But it is very important for us to have an accurate target variable and thus the estimated losses cannot be used directly as our target variable for the pattern recognition. Here we will show some statistical results for the estimated actual losses, which are calculated by the first method. Later in the next section we will show how we reduce the effect of the estimation error on our analysis based on the results presented here. Table 3.2 below shows a brief summary for losses happened to all companies across all contract years. Unit: U.S.Dollars.

Table 3.2: Statistics for Loss

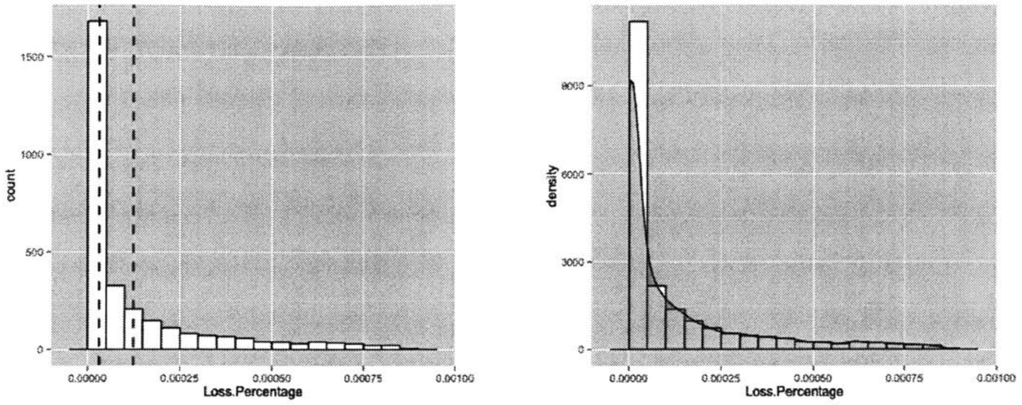
Minimum	First Quantile	Median	Third Quantile	Maximum	Mean
0	2417	36540	479200	373800000	3101000

We can see that the range of values between and within each quantiles are very large. The phenomena is understandable because the clients have very different market caps and the amount of loss depends on the size of companies. Thus it is more meaningful for us to look at what we called the loss percentage, which is defined as $\frac{\text{Total amount of loss}}{\text{Total Insured Value}}$. Again, we show the basic statistics in the following Table 3.3.

Table 3.3: Statistics for Loss Percentage

Minimum	First Quantile	Median	Third Quantile	Maximum	Mean
0	0.0000033	0.0000623	0.0003719	0.6521000	0.0012890

Although the values are not as large as those in loss amounts, the ranges are still very large both for between and within intervals. The average percentage of losses is around 0.1 percent and the median is 20 times smaller, indicating a right-skewed distribution. Indeed, a more detail exploration shows that 3061 out of 3535 contracts have loss percentages less than 0.1 percent. The mean value is pulled to the right by those rare but relatively large values. In order to visualize the overall distribution, we ignore temporarily for now the very large values and focus only on the first 3000 loss values.



(a) Histogram:Loss Percentage

(b) Density plot of Loss Percentage

Figure 3-1: Distribution of Loss Percentage

Figure 3-1 above shows the distribution of loss percentage of 3000 contracts. The histogram in figure (a) shows the frequency distribution of loss percentage; the blue line represents the median while the red line represents the mean value. Figure (b) provides an estimated density plot over the histogram. Both figures indicate that loss percentage is highly right-skewed and the mean value is affected by those extreme values at the tail. If we add the left 535 extreme values in, the mean will be pulled to the right even more. We observe that the tail of the density plot starts approximately from 1×10^-4 and all values larger than it are components that more or less drag the mean to the right. In the normal boxplot, these values are treated as outliers and we should separate them from the normal values. Moreover, we also notice that there are 466 contracts with zero loss. These contracts are obviously different from all others and they represent a group of safe contracts; we should also separate them from others as well.

Except for those on the very left side with zero loss and the values on the tail, all others are concentrated in the subinterval at the left of the axis. Based on all of this information, we come up with the idea to create buckets based on these subintervals and put together clients with loss percentages lying in the same areas. In next section, we will use this idea to construct our target variable in pattern recognition and this will be the most important preprocessing technique involved in our research.

3.3.2 Impact of Industry

We mentioned at the very beginning of this thesis that one of the challenges insurance companies are now facing is the diversity of their clients. Our collaborator is now providing products to clients across hundreds of industries and the variety makes their risk assessment harder than ever before. More importantly, based on common knowledge and their experience in insurance, industry has a huge impact on clients' potential risk. The logic is quiet intuitive; we will nearly always expect a manufacturing company running hundreds of factories producing combustible materials such as woods to be more risky than a technology firm selling software products to customers through online access. Although we all know industry should be an important factor in determining the potential risk levels, the problem is how much impact it has on our analysis.

In the database, the only attribute related to industry is the Standard Industrial Code (SIC) and our analysis is based on it. Standard Industrial Code (SIC) is a four-digit numerical code assigned to companies, representing their primary business. Ideally if we have access to a more comprehensive database, we can calculate risk score for each industry based on the performance of the enterprises lying in each industry. However, we are restricted to the internal database and analysis can only be conducted on the relatively small pool. Fortunately, we are still able to get some insights in quantifying the financial risk hidden in each industry based on the statistical analysis shown below.

Based on our knowledge of SIC, the first two digits of the code identify the major industry group and the last two digits represent more detailed categories. With a larger data sample, we might be able to work with a whole SIC list; however, due to the restriction of sample size, we only focus on the first two digits of the code. Table 3.4 below provides the ten major industry groups represented by the first two digits of the code. We can see that the list is inclusive and nearly all industries are included, from those with relatively low risk exposure such as Services to those with relatively high risk exposure such as Mining and Construction.

Table 3.4: Major Industry Group Represented by SIC

01-09	Agriculture, Forestry, Fishing
10-14	Mining
15-17	Construction
20-39	Manufacturing
40-49	Transportation & Public Utilities
50-51	Wholesale Trade
52-59	Retail Trade
60-67	Finance, Insurance, Real Estate
70-89	Services
91-99	Public Administration

As we have shown previously, we have in total of 3035 clients. We observe that these clients cover all ten major industry groups listed above and more precisely, they cover 73 different industries. The pie chart in Figure 3-2 shows the distribution of clients in major industry groups and we can see that over 50 percents of the clients are in the field of manufacturing while wholesale trade, as the fifth occupied industry, only have around 6 percents of the client lying in the field.

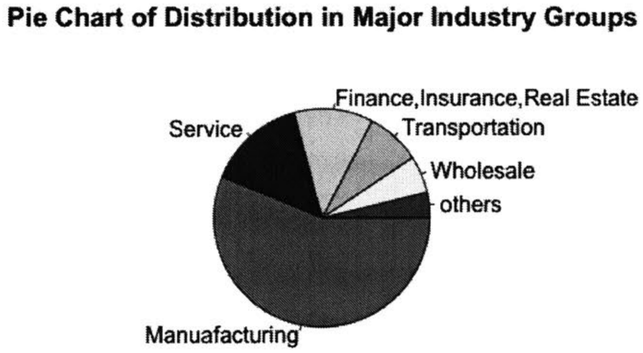


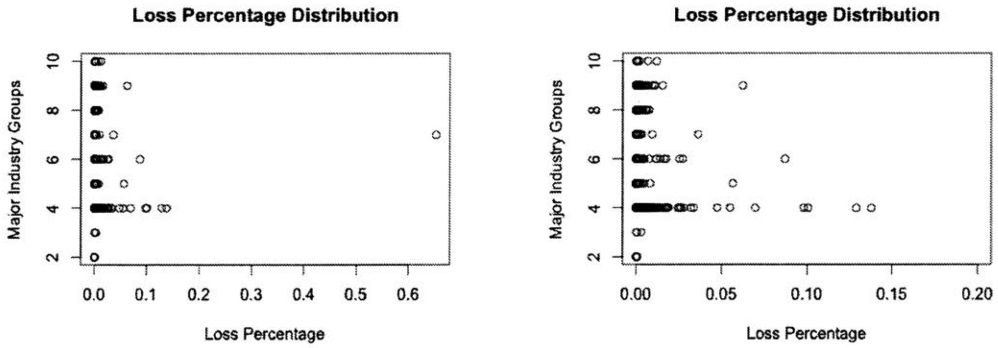
Figure 3-2: Pie Chart of Five Most Occupied Major Industry Groups

Although clients seem to concentrate in a major industry group, they are indeed widespread in many different minor industries. Table 3.5 shows the five most occupied industry groups and the number of clients in each group. We notice that the clients are indeed very widespread in all industries and there are only 8.8 percents of clients in the most occupied industry.

Table 3.5: Five Most Occupied Industry Groups

Code	Count	Industry Group
36	170	Electric & Electrical Equipment
28	179	Chemical & Allied Products
20	210	Food & Kindred Products
35	212	Industrial & Commercial Machinery
65	268	Real Estate

Next we try to build up a direct relationship between major industry groups and financial loss and indeed we find that clients in certain groups perform differently with other clients in terms of financial loss. The plots in Figure 3-3 show the distribution of loss percentage for clients across the ten major industry groups; the left one shows the whole distribution and the right one shows the distribution without outliers. The numbers on the vertical axis represent different industry groups and the values represent the value of the first two digits of SIC. For example, 4 represent the fourth largest two digits SIC, which ranges from 20 to 39.



(a) Complete Distribution

(b) Distribution without Outliers

Figure 3-3: Loss Percentage Distribution for Clients Across Industries

We observe that the fourth group perform very differently from all other groups.

While most of other groups have dots concentrated in the left, the fourth group has a widespread distribution, meaning that clients in this group tend to perform different from each other in the group. We also notice that all other groups perform very similarly except for a few outliers. The observation enlightens us to separate the fourth group with all others. We check that clients in the fourth group has SIC starting from 20 to 39 and the major industry represented is manufacturing. Previously, we have already shown that manufacturing is different from all others and it has more than half of clients lying in the field. This further strengthens our confidence to make the separation. In the next section, we will show how we make use of this conclusion to create the industrial feature as one of the input variables.

3.3.3 Original Risk Assessment

We have already discussed that the original risk system can no longer satisfy our collaborator. Although it is claimed to be an inefficient and ineffective system, we still feel it necessary to discuss it in more detail because the risk score will be used as one of the reference when we conduct comparison in the next chapter. The original risk score is called the risk quality rating (RQR) and the score is a weighted average of engineering scores. Here we will show some basic statistics about the score and conduct a minor comparison between the score and loss percentage which is directly related to financial loss.

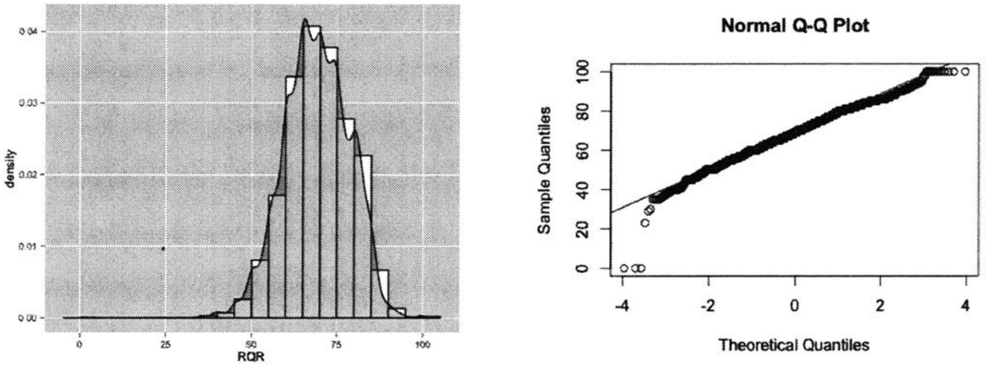
Risk Quality Rating (RQR) is a score ranging from 0 to 100 and in principle, the lower the score is, the more risky a client will be. Table 3.6 shows the most basic statistics for the score recorded in our database. We can see that the overall score is not bad with mean and median close to each other. The score might have a nice symmetric distribution.

Table 3.6: Statistics for Risk Quality Rating

Minimum	First Quantile	Median	Third Quantile	Maximum	Mean
0	63	69	76	100	68.95

Indeed, our intuition is right and the score does have a symmetric distribution.

Further test shows that the distribution is very close to a normal distribution. This is understandable because the Risk Quality Rating itself is a score calculated by a system which uses the weighted average method. In other words, the score consists of engineering scores which are designed to be normally distributed. Although the weighting might destroy some normality, RQR should still have a distribution similar to the normal. The plots in Figure 3-4 validate our inference. The left one is the histogram with density curve on top and it shows the overall distribution; the right one is the normal quantile-quantile plot with the Q-Q line on it. We can see that the overall shape is similar to the the bell-shape and the points in the Q-Q plot are mostly located on the Q-Q line which is in red. All the observations are evidence that lead to a normal distribution.



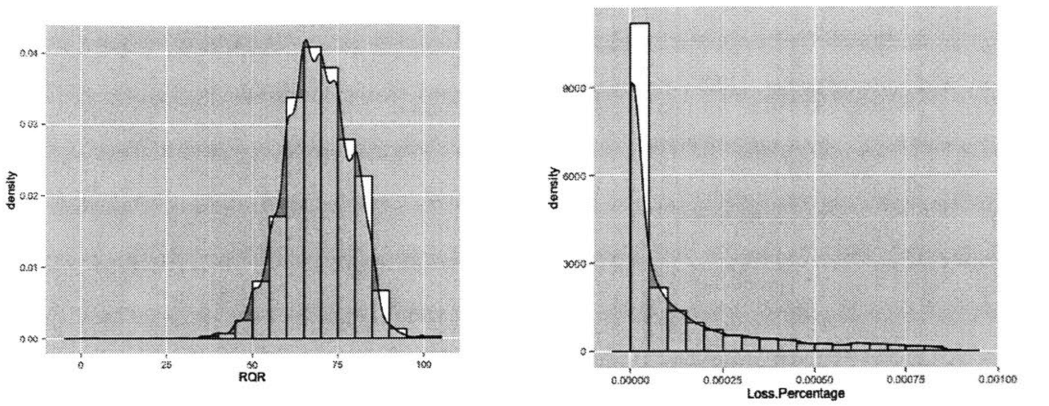
(a) Histogram:Risk Quality Rating

(b) Normal Quantile-Quantile Plot: RQR

Figure 3-4: Distribution of Risk Quality Rating

However, as we have stated in Section 3.3.1, the distribution of loss percentage is highly right-skewed. The inconsistency between the score and truth arises and the distributions shown in Figure 3-5 provide a strong comparison. Figure 3-5a shows a symmetric bell-shaped distribution of the original risk score, meaning most of the clients are around the average risk level; Figure 3-5b is the actual highly right-skewed distribution of loss percentage, which is directly related to client financial loss. We observe two totally different distributions and thus get to the conclusion that the original risk score, the Risk Quality Rating, cannot represent the actual loss pattern for clients. Later in the next chapter we will show another comparison between Risk

Quality Rating and financial loss in a different perspective.



(a) Distribution of Risk Quality Rating

(b) Distribution of Loss Percentage

Figure 3-5: Comparison between RQR and Loss Percentage

3.4 Feature Construction

The purpose of applying machine learning techniques is to uncover information hidden in our datasets; thus the whole process can be viewed as an exploration of the hidden relationship between the input variables. The input variable, or feature, as defined by Bishop (2006), is a measurable property of a phenomenon being observed. Bishop (2006) also points out that choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, classification and regression.

Originally, features could be numeric, structural, or categorical. Some of them could be redundant, or too large to be managed or not directly applicable for existing models. Therefore, a preliminary step is needed before we can assemble our input matrices and we call this step feature construction. This step is crucial to many machine learning applications and could potentially lead to the success or failure of the whole project. Feature construction includes selecting a subset of existing variables and transforming part of the selected set to a new space where the problem solving can be easier. Another purpose of feature construction is to improve computation efficiency; original variables or raw data could be of any type and direct application could lead to a very slow and inefficient algorithm.

In this section, we will discuss the key feature construction process involved in our research and we will show how the process help with our analysis. The construction is based on the preliminary analysis shown in the previous section and also based on discussions with our collaborators.

3.4.1 Time line

Before we go deeper to feature construction, it is important to discuss about the time line. Time is important for machine learning projects. We are using history to predict future and thus it is extremely important to define at the very beginning what do we mean by history and future. In this section, we will discuss the time line we set for our research; specifically, we will show how we define history, current, and future. As we have stated earlier in the chapter, data provided to us are from 2008 to 2013 and different parts of data are recorded in separate files. A major problem here is that different files are recorded by different departments, causing inconsistency in time frame. For example, we have the contract information for a client from 2008 to 2011 but the claims are only recorded in 2009 and 2011. Indeed, the problem is common in data mining and it is always critical for data scientists to explore a suitable and reasonable way to solve the problem. This is what we call missing value imputation in machine learning and data mining. Fortunately, the problem is also a commonly happened issue for insurance companies and they have already come up with a solution based on their experience. Our collaborators suggest to let contract prevail in case of any inconsistency between the claims and contract. For each client, we take its contract information and then refer to the corresponding claim record. If there is any contract year without claim record, we then add the claim record for those years as zero. Although we believe there should be a more rigorous solution, this is all we can do based on what we are provided and nearly every missing value imputation is not perfect.

Although there are in total of 3035 clients on record, not all of them have same

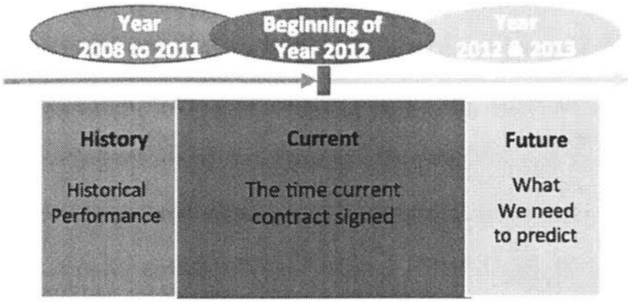


Figure 3-6: Time Frame

contract years. We might have contract information for client A from 2008 to 2010 but from 2009 to 2012 for client B. In order to maximize the number of clients for our analysis, we define the year with the most clients in contract to be the current time. According to statistics, there are 1732 clients with contract in year 2012 and this is the most we can have. Thus everything before the beginning of 2012 is defined as history and what happened in and after 2012 are all future informations. At this point, we are using information prior to 2012 to predict situation in 2012 and 2013. The figure above gives a better visualization of the time frame and our analysis is conducted accordingly.

We believe that a fixed time line better than a relative concept of history and future. We could define history and future differently for each client; just to set the newest contract year as future, the beginning of the year as current, and everything happened before the year as history. However, the problem here is that we have to perform missing value imputation for each client. If we define a relative time line, each client might have different years of historical information, which leads to an unfair situation.

3.4.2 Target Variables

A target variable is nearly always one that attracts the most concern simply because that is exactly what people want to know. As Abbott (2014) states that a target variable carries with it all the information that summarizes the outcome we would like to predict from the perspective of the algorithms we use to build the predictive models. Abbott (2014) argues that defining a target variable is the most critical step in the process that relates to the data, and more important than data preparation, missing value imputation, and the algorithm that is used to build models, as important as they all are. Although a machine learning or data mining project would not succeed without any of the steps mentioned above, we have to admit that every project has to begin with defining what problem will be solved. We need to know clearly at the very beginning that what we want and everything else is designed to achieve the goal.

In this research, we have two clear goals: to recognize client loss patterns and to predict whether a specific contract would be profitable (gain) or unprofitable (loss). Thus we need to define two target variables, one for each goal. For the gain-loss classification, defining the target variable is quiet straight forward. The outcome of this model is obviously binary and thus we define 0 as profitable (gain) and 1 as unprofitable (loss). The problem here is to define 0 and 1 and our collaborator provided us with the criterion. According to their expertise, if the ratio of loss to premium is greater than 0.35, the client would be unprofitable. Thus our target variable is defined as follows.

$$\text{Target Variable} = \begin{cases} 1 \text{ (Unprofitable)}, & \text{if } \frac{\text{Amount of loss}}{\text{Premium}} > 0.35. \\ 0 \text{ (Profitable)}, & \text{otherwise.} \end{cases} \tag{3.1}$$

Next we will define the second target variable for loss pattern recognition. As we mentioned earlier in the preliminary analysis that the distribution of loss percentage is highly right-skewed and the tail of the density plot starts approximately from $1 \times 10^{-}4$. Moreover, there are 466 contracts with zero loss. These two groups of contracts are obviously very different from all other contracts and we want to put them into separate groups. The idea is to create buckets based on the three subintervals and put together clients with loss percentages lying in the same areas. Thus accordingly, we define a new variable called loss percentage level. Loss percentage level, as shown by its name, is a discrete variable with three different levels. The levels are converted from the continuous variable loss percentage according to Table 3.7 below.

Table 3.7: Conversion Table for Loss Percentage Level

Loss Percentage Level	0	1	2
Loss Percentage	0	$[0, 1 \times 10^{-}4]$	$[1 \times 10^{-}4, \infty)$

Till now, we have created two variables as target variables for our analysis. The binary variable for gain-loss classification and the discrete loss percentage levels for the loss pattern recognition. In chapter 4, we will show the modeling process for whose output is loss percentage levels and in chapter 5 we will propose our own algorithm whose objective is to predict correctly the binary label for each clients.

3.4.3 Input Variables

Input variables, seemingly not as important as target variables, are indeed critical for analysis. The logic is straightforward; it is impossible to make predictions without inputs. Target variables as outputs are what we want to know and input variables as inputs are what we already know however not completely. There is information hidden in input variables and our research is to uncover as much of the information as possible. In this section, we will discuss how we create some of the important input variables from the database and later in the next section we will provide a full list of variables used in our analysis.

In the previous section, we discuss the impact of industry and point out that clients with Standard Industry Code (SIC) starting from 20 to 39, major industry group manufacturing, are different from all other clients in terms of financial loss distributions. Also manufacturing has more than half of clients lying in the field. Both observations lead us to the same conclusion, to separate clients in manufacturing from other clients. Thus we create a binary variable as one of the input to represent industry. Specifically as follows.

$$\text{Industry} = \begin{cases} 1, & \text{if client belongs to manufacturing.} \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

Originally, industry is a categorical variable represented by SIC. As we have shown previously, SIC is a four digit code with thousands of different values. The raw variable is obviously too large for to use and we will never want to use this kind of variable in our analysis. Therefore, based on our observations we transform the original variable into a new feature space where it only takes two values. It is still categorical but much easier to use. In this way, we greatly improve the computation efficiency and at the same time keep as much of its original information as possible.

Similar to industry, we also create a binary categorical variable for location. For insurance, domestic accounts and international accounts are totally different. Thus we put all accounts in the State as a group and all others including accounts in Asia, Europe, and all other countries as the other group. The variable is created according to the following rule. Similarly, a categorical variable is again transformed to a smaller feature space, reducing its variability but still informative.

$$\text{Input Variable: Region} = \begin{cases} 1, & \text{if account is in the State.} \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

An important part of input is historical performance. This informations is so important that sometimes people might even naively make conclusion solely based on history; they claim that "How you perform historically determines how you will perform in the future." Although the statement is naive and somehow arbitrary, there is still something to recommend it. In our analysis, we create two input variables according to history. The first one corresponds to one of the target variables, the loss percentage levels. We take the average of loss over all historical contract years and calculate the average historical loss percentage level as an input. Additionally, we create the second variable, the historical loss frequency; that is, we take take the average number of claims over all historical contract years as another input variable. The motivation here is intuitive and the inspiration comes from personal lines. A person who reports many claims of his car in the history might be not as cautious as those who report less claims and thus this car owner might keep reporting more claims in the future because of his recklessness. Similarly, a company which reports a lot of claims in history might because of the nature of its business or because of its employees' recklessness and thus this company will have a large possibility of reporting more claims and more loss in the future because of all these unchanged factors.

Loss can be labeled as catastrophic loss and non-catastrophic loss according to its causes. The classification is important to the insurance industry because different category of loss might lead to different claim policies. Thus we calculate the percentage of loss due to catastrophe for each client and use it as another input variable. This variable is continuous and easy to be used directly and thus we do not conduct any further preprocessing on this variable.

Till now, we have presented all important input variables and the preprocessing techniques and missing value imputation we use for each of them. The preprocessing is conducted partly based on our preliminary statistical analysis and also based on suggestions from experts in the field of insurance. Although we have achieved satisfying results based on the preprocessing, we are not sure whether the techniques will work for other insurance database and there might be better techniques for dataset with different features. In the next section, we will provide a full list of variables, including all input and output variables.

3.4.4 A Full List of Variables

In this section, we summarize the chapter by a comprehensive list of variables. We provide the name of the variable and a brief description associated with each of them. Variables listed in Table3.8 attached in the following page include important input and output variables involved in this research. As we pointed out earlier, we have two models thus two target variables and they are named as "Future Profit label" and "Future Loss Percentage Level" in the table respectively. Notice that there are two input variables with name "historical profit label" and "historical loss percentage level"; these two variables have the same definition with the target variables but in different time frame. However, Loss Frequency is only defined for historical contract years and we do not involve future loss frequency into analysis.

Table 3.8: List of Variables in Analysis

Name	Description
Reference Number	The internal index assigned to the client
Region	Binary Variable: Domestic (0) or International (1)
Total Insured Value (TIV)	Numeric:The total amount insured for the client
Property Damage Percentage	Numeric:Percentage of damage to property caused either by people other than its owner or by natural disaster
Business Interruption Percentage	Numeric: Percentage of the loss of income that a business suffers after a disaster
Historical Loss Percentage Level	Discrete input variable, showing the level of historical loss percentage. Value takes 0, 1, or 2
Future Loss Percentage Level	Discrete target variable, showing the level of future loss percentage. Value takes 0, 1, or 2
Standard Industrial Code (SIC)	Binary Variable: Manufacturing (1) or Other industries (1)
Share percentage	Numeric: The percentage that the insurance company is responsible for loss incurred by the client
Risk Quality Rating (RQR)	Numeric: Risk score from 0 to 100, showing the risk level of the client; also the original risk index used by the company
Catastrophe loss Percentage	Numeric: Percentage of loss due to catastrophe, such as earthquake or flood
Historical Profit Label	Binary Input Variable: (0) Profitable or (1) Unprofitable; preprocessed from ratio between historical loss and premium, threshold: 0.35
Future Profit Label	Binary Target Variable: (0) Profitable or (1) Unprofitable; preprocessed from ratio between future loss and premium, threshold: 0.35
Loss Frequency	Numeric: Average number of claim happened in historical contract years

Chapter 4

Client Loss Pattern Recognition

4.1 Introduction

In this chapter, we focus on the first part of our research, client loss pattern recognition. As stated earlier at the beginning of this thesis, capturing clients' loss pattern is very meaningful for insurance companies. It will help them better understand the impacts of industry, location, and other features that would affect the claim amount of their clients. Previously we showed in Chapter 3 that the original risk rating system designed by our collaborator is not able to represent the true loss pattern of their clients and thus a new system is in need. In this chapter, we will discuss about the reference system we build and we will focus on the associated risk index calculated by the system. In Section 4.2 we will introduce the new risk index, its definition, method of calculation and the theoretical basis associated with the index. Later in Section 4.2.3 we proceed to present results from numerical experiments, showing the power of the new risk system by comparisons between the distribution of actual loss and our index. We also compare the performance of the new index and the original risk score, the Risk Quality Rating. Finally, we will summarize the chapter in Section 4.3.

4.2 A New Risk System

The traditional risk assessing system is based on the on-site engineering investigations; ideally, if an insurance company can afford comprehensive and inclusive investigations, engineers would be able to assemble a large database to record detailed information for factories, branch offices, and everything related to their clients. Then analysts should be able to calculate accurate risk scores for their clients across all industries around the world. However, in real world this greedy investigation is extremely costly and no insurer would like to spend so much just for risk assessment. Thus the resulting database can only include a certain portion of the information needed for such analysis. In view of the strict constraint, we decide to build a new risk reference system based on a different database. Instead of relying on on-site investigations, we focus on insurer's internal business-related database which can be generated easily from clients' profile and other public data source. Examples of attributes include major industry group and total insured value. This is an important innovation in the field of insurance and to the best of our knowledge, there has been no risk assessing system based solely on business related data. With the new reference system, risk assessments are no longer constrained by the limited amount of engineering data and thus we can potentially generate more accurate risk representations. Moreover, the system provides a much more efficient and effective risk assessing procedure and thus assist insurers with cost control by saving the huge expense on engineering investigations.

In this section, we will discuss the risk reference system in detail and especially, we will focus on the output of the system, the risk index. In Section 4.2.1 we will first define the index, then show the model of calculation, and finally provide the theories that support the system. Later in Section 4.2.3 we will show results from numerical experiments and compare the performance of the new risk index and the original index. With plots and comparisons, we will show how exactly the reference system captures the loss pattern.

4.2.1 The New Risk Index

The risk index is the output of our innovative risk reference system and our objective here is to capture the financial loss pattern for all clients. Thus the index should be built upon the attributes that are directly related to financial loss. According to our descriptions in Chapter 3, the variable should be the loss percentage level. Moreover, time line has already been defined and we are trying to use current and historical information to predict future. Thus the new risk index would naturally be defined as the future loss percentage level; however, we would never be able to know what would happen in the future and thus our index can only be calculated as the expected value of future loss percentage level.

All above statements lead us to the final definition for our new risk index: the expected future loss percentage level. Indeed, the index is a weighted sum of future loss percentage level and here we transform the nominal variable loss percentage levels to a numeric variable. But the transformation is quiet reasonable since higher value of loss percentage does represent a higher amount of loss. The following formula gives a direct and more concise definition of the risk index. Risk Index = $\sum(\text{future loss percentage level} \times \text{associated weight})$. We can see that the range of the index is still $[0, 2]$. Unlike the loss percentage level, which is a discrete nominal variable, the new risk index becomes a continuous numeric variable.

We first approached the problem as a multi-class classification with target variable being future loss percentage level. However, as is well known, multi-class classification problems are harder than binary problems and it would require much efforts to get accurate results. Thus we come up with a solution to tackle the problem from a different perspective; instead of taking directly the label assigned by the classification algorithm as the final output, we take intermediate results from the model as weight values. Specifically, we first define the future loss percentage level as the target variable for a probabilistic classifier and train the algorithm; we then take the probabilities for each data points being in each loss percentage level as the weight factor. Finally we multiply the weights by the associated levels to get the expected value as our index. The flow chart in figure 4-1 summarizes the calculation process and it better explains the function of classifier in the process.

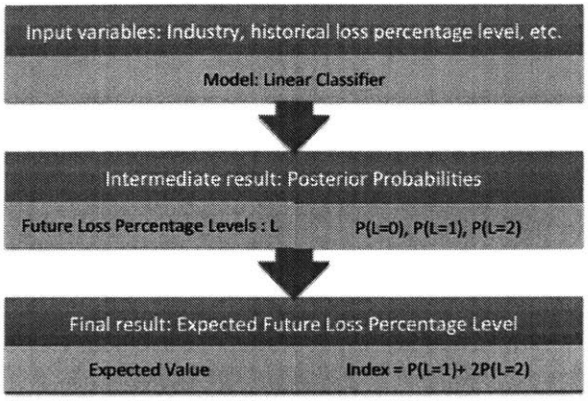

Input variables: Industry, historical loss percentage level, etc.
Model: Linear Classifier
Intermediate result: Posterior Probabilities
Future Loss Percentage Levels : L
P(L=0), P(L=1), P(L=2)
Final result: Expected Future Loss Percentage Level
Expected Value
Index = P(L=1)+ 2P(L=2)

Figure 4-1: Flow Chart: Calculation Process for New Risk Index

The new risk index could potentially reflect the risk level for clients. However, it is important to note that this index is a relative concept and it is not directly related to what we define in the previous chapters. The index, despite calculated as the expected future loss percentage level, is the risk assessment in a new risk system and thus cannot be used directly as the loss percentage level. It is a relative concept and shows only the relative risk level for clients. In other words, a client with index value 1.5 does not mean its future loss percentage level is 1.5; it means that this specific customer is less risky than clients with index value larger than 1.5 and more risky than clients with index value smaller than 1.5.

4.2.2 Probabilistic Classifier

In the previous section, we briefly introduced the calculation process for our risk index. In this section, we will focus on the calculation of weights, show the model we use and provide theoretical explanations for the choice. The calculation of weights is obviously the most important step in our calculation process and it is directly related to the accuracy of our index. The use of a probabilistic classifier is straight forward since what we actually need is the probabilities but we need to choose from different classifiers in the group.

Among all probabilistic classifiers, Linear Discriminant Analysis and Logistic Regression are two of the most popular ones. Linear Discriminant Analysis as generative model and Logistic Regression as discriminative model are widely used in many different areas; these two models can be extended to multi-class classification with easy revision on the original versions. Thus we first narrow our choice to be either one of the two. One important difference between generative models and discriminative model is that generative models are built upon the joint probabilities of class labels and input variables while discriminative models are based on the conditional probabilities of class labels provided input variables. Jordan (2002) compares generative learning and discriminative learning using Naive Bayes and Logistic Regression as examples.

Linear Discriminant Analysis (LDA) is a commonly used technique in machine learning and pattern recognition. Its main objective is to reduce dimensionality while preserving as much of the class discriminatory information as possible and thus it is in many cases closely related to Principle Component Analysis (PCA). Since it preserves the class discriminatory information, it can also be used as a classification model to classify items into different categories. Logistic Regression, despite its name, is a type of discriminative classifier built upon the conditional probabilities of data points being in a specific class provided the input variables. It is closely related to the exponential families and uses log-odds in the model set up. In our case, what we concern the most are the conditional probabilities, which we will use as the weights. In other words, we want to know what is the probability of data points being in the three loss percentage levels given all their features. Thus for our case, if we choose Logistic Regression, we can get what we want directly. However, if we choose Linear Discriminant Analysis, we would need to calculate the weights from joint probabilities with Bayes Rule.

We denote the input features as X, the class as G; X will be a vector in multidimensional space and G will be the future loss percentage level belonging to the set $0, 1, 2$. For Logistic Regression, the conditional probabilities could be calculated by equation 4.1.

$$P(G = k|X = x) = \frac{e^{\beta_k \times x}}{1 + \sum_{i=0}^{1} e^{\beta_i \times x}} \tag{4.1}$$

Notice that $k = 2$ is chosen to be the pivot class and the associated probabilities can be calculated by $P(G = 2|X = x) = 1 - P(G = 0|X = x) - P(G = 1|X = x)$. The parameters can be estimated by maximum likelihood and calculations are straight forward.

However, for Linear Discriminant Analysis, the calculations are not as easy. If the prior probabilities of class k are π_k, and class-conditional densities of X in class $G = k$ are $f_k(x)$. The posterior probability of x being in k, $P(G = k|X = x), k = 0, 1, 2$ can be calculated according to equation 4.2.

$$P(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{i=0}^{2} f_i(x)\pi_i} \tag{4.2}$$

From the calculation perspective, we would prefer Logistic Regression because of its simplicity. Indeed, we do choose Logistic Regression to calculate the weights but there are more important reasons, which we will discuss in detail now. The most important unknown in equation 4.2 is the term $f_k(x)$, which corresponds to the class-conditional density function of X. This is where the assumption of Linear Discriminant Analysis (LDA) arises. In LDA $f_k(x)$ is assumed to be multivariate Gaussian and it also assumes that all classes share a common covariance matrix. Friedman et al. (2001) lists the expression of $f_k(x)$ as in equation 4.3, where Σ represents the covariance matrix and μ_k is the mean of X in each class.

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)} \tag{4.3}$$

Again, we go back to the important assumption in Linear Discriminant Analysis. It assumes that the class-conditional density are multivariate Gaussian. However, the assumption does not hold true in some cases since the assumption seems to be too strong. As we can see in Chapter 3, there are a certain number of qualitative or nominal variables in our case and thus the multivariate normality would not happen. This directly breaks the assumptions of Linear Discriminant Analysis and thus the result from LDA might be really bad. Thus we finally choose Logistic Regression to calculate our risk index.

Jordan (2002) and Press and Wilson (1978) discussed Logistic Regression and Linear Discriminant Analysis and provided comparisons. Press and Wilson (1978) point out that if the populations are normal with identical covariance matrices, Linear Discriminant Analysis is likely to give better results. However, if normality does not hold true, logistic regression would be a better choice. In fact, Logistic Regression, as a newer and more generally method of classification, is more robust. However, it does not mean that Logistic Regression is a perfect estimator and it does have its own limitations. As pointed out by Elkan (2013), a major limitation is that the probability as a function of each predictor must be monotonic. Thus instead of setting assumptions on the distributions of indicators, Logistic Regression puts limitations on the probability function. Later in Section 4.2.3, we will apply both Linear Discriminant Analysis and Logistic Regression to our dataset and compare their performances in capturing the loss patterns. Indeed, we will see that the results do not differ as much as we expect and Linear Discriminant Analysis nearly perform as well.

4.2.3 Numerical Experiments

In this section, we will present results from numerical experiments, showing the ability of our new risk system to capture the financial loss pattern. Results from Logistic Regression and Linear Discriminant Analysis will be compared later and we will also conduct several comparisons between our new risk index and the Risk Quality Rating. we will start with discussion about data division.

As we mention in Chapter 3, this part of research is conducted on the client level and we want to capture the loss pattern for clients not for contracts. As we stated earlier, there are in total of 1732 clients available for our analysis. We divide all these clients into the training and test sets by the ratio 2 to 1. Specifically, the number of clients in training set is twice as many as the number of clients in test set. In practice, we first generate $1732 \times \frac{2}{3} \approx 1154$ indexes from 1 to 1732. We then assign data points with these indexes to be in the training set and the rest will be assigned to be test set. We then train our classifiers with the training dataset, apply the trained algorithms to the test set to get the posterior probabilities, and finally calculate the expected future loss percentage level as the value for the new risk index for all clients in the test set.

We will first show how the new risk index captures the financial loss pattern by comparing the distribution of the new index to the actual future loss percentage levels. We first apply our trained algorithm to the test set, calculate the values of new risk index for all clients in the set, and plot three boxplot for data points with actual loss level from 0 to 2. The three boxplots in Figure 4-2 present the distributions of our new index versus actual future loss percentage levels. The horizontal axis is the actual loss percentage levels and the boxplot above each level are the corresponding distribution of our new index.

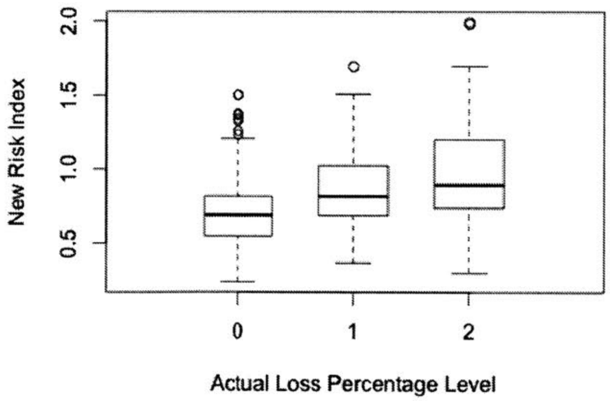


Figure 4-2: Distribution of New Risk Index vs. Actual Loss Level

We observe that as the actual loss percentage level moves from left to right, the corresponding boxplot moves up accordingly and all key attributes of the boxplot will increase. We see even the outliers would have larger index value when the corresponding actual loss gets higher. On the other hand, the original risk score, Risk Quality Rating, does not work well. As it is designed to be, Risk Quality Rating should be lower for clients with high loss. However, as we can see from figure 4-3 that distribution of RQR does not show any obvious pattern with the change of actual loss level. Indeed, when actual loss level changes from 0 to 1, the boxplot of RQR moves to the opposite direction.

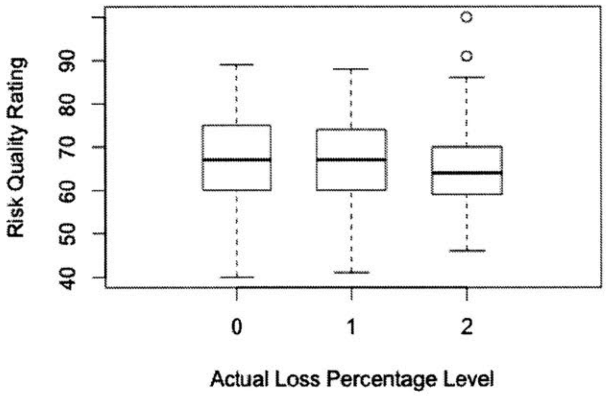
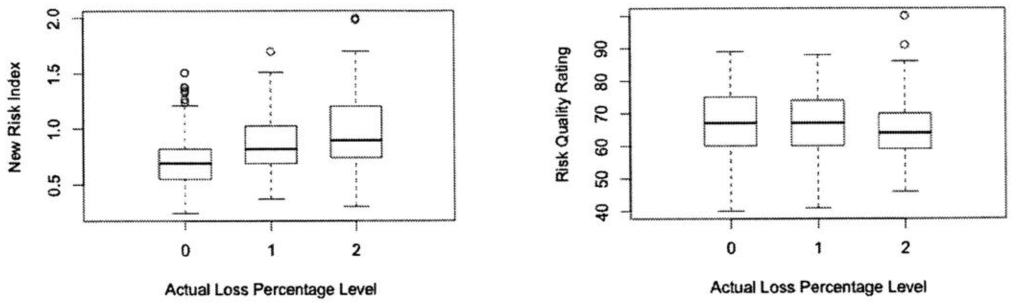


Figure 4-3: Distribution of Risk Quality Rating vs. Actual Loss Level

In figure 4-4, we put the above two figures side by side and we can have a better visualization on the power of our index. The observation shows that change of the new risk index is in accordance with the change of the actual loss while the original index changes more randomly. This leads us to the conclusion that our new risk index better captures the pattern of actual loss and this fulfills our first objective in this research.

As we discuss earlier in the previous section, we have been struggling to choose from Linear Discriminant Analysis and Logistic Regression as the classifier for our project. Although we come up with a strong reason of not using Linear Discriminant Analysis, we are still interested in its performance. Figure 4-5 shows the same boxplot



(a) Boxplot: New Risk Index (b) Boxplot: Risk Quality Rating

Figure 4-4: Comparison: New Risk Index vs.Risk Quality Rating

for the new risk index calculated by Linear Discriminant Analysis. Indeed, we have to admit that the algorithms works nearly as well as Logistic Regression; however, the variances seems to be larger and we observe more outliers.

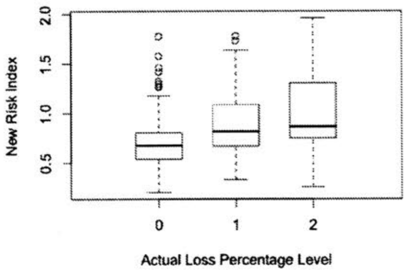
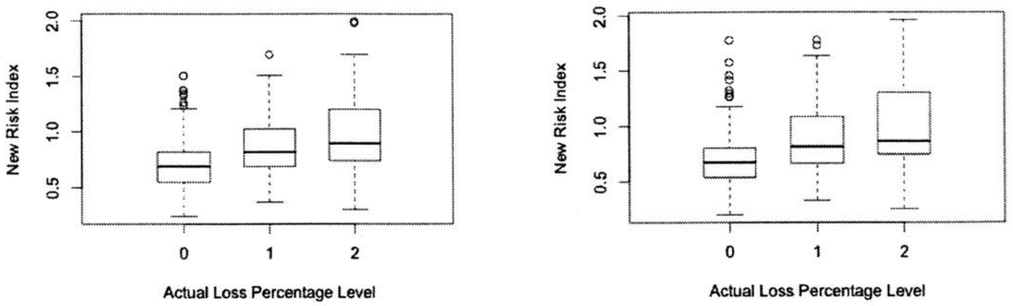


Figure 4-5: Distribution of Risk Quality Rating vs. Actual Loss Level

These are minor differences and they are not so obvious. Figures 4-6 can help us to see the difference better. We can see that boxplot corresponding to Linear Discriminant Analysis have whiskers extending further to the end and there are more dots outside of the end of the whiskers. Although we only observe minor differences in our case, as database gets larger and more comprehensive, the violation of the assumptions in Linear Discriminant Analysis might have a larger impact on the result and the difference might be much more obvious.

Indeed, Press and Wilson (1978) conduct some numerical experiments and compares the performance of Logistic Regression and Linear Discriminant Analysis and



(a) Boxplot: New Risk Index by Logistic Regression

(b) Boxplot: New Risk Index by Linear Discriminant Analysis

Figure 4-6: Comparison: Linear Discriminant Analysis vs. Logistic Regression

they conclude that tt is unlikely that the two methods will give substantially different results, even though the assumptions are slightly violated. However, as we have stated above, we cannot foresee the potential impact the assumptions would have to the index as more data are included and thus we still choose Logistic Regression as the only classifier in our model. In fact it is better both theoretically and practically.

4.3 Conclusion

In this section, we conclude our findings and results in the first part of our research, where we strive to uncover the financial loss pattern for clients. We build a new risk assessing system and the key of the system is to develop a new risk index. The index is the expected value of the future loss percentage level and it is calculated as a weighted sum of future loss levels. The associated weights are the conditional probabilities for each data point being each loss level. The probabilities are calculated by probabilistic classifier and our analysis shows that Logistic Regression as a discriminative model would be a better choice than Linear Discriminant Analysis as a generative model.

Results form numerical experiments show that the new risk index changes in accordance with the actual loss level while the original risk score, the Risk Quality Rating, changes randomly. Thus we conclude that the new system captures the financial loss pattern for clients across all industries around the world and it provides a better reference to the insurers than the original risk score. Further, by completely putting aside engineering investigations and focusing only on internal business data, we provides a totally new perspective of risk assessment. They system we built provide a more efficient risk assessment method with extremely low cost; instead of spending huge amount of money hiring engineers to perform several months of investigations, our method only requires a few days of data collection, preprocessing, and modeling. In our case, it takes our machine less than an hour for the modeling process. As database gets larger, it might take longer but time will always be reasonable and much less than the original process. Although more work is needed to be done to improve the performance of our system, we still gives a more accurate assessment with lower cost in less time. We believe it might be the first step to break the tradition of risk assessment in insurance.

Chapter 5

Profitability Prediction

5.1 Introduction and Motivation

As for-profit organizations, companies are most concerned about profit and profitability is the most concern for their executive teams. The rule applies to insurance companies; when a new client comes to them, the first thing the insurance company would like to know about is that whether this specific contract with the client can bring profit to his company. Before a undertaker signs a contract, he really wants to make sure that the contract in front of him would not lead to huge amount of claims. Unfortunately, as far as we know, there has not been such a system available to help the insurance companies with decision making. This motivates us to build the system that can label the contracts for our project collaborator as profitable or unprofitable. If the system can provide relatively accurate labels, it will potentially be able to bring huge profits to the insurance companies and help them expand the market with acceptable risk.

As we can see from Figure 5-1, there are much more profitable (0) contracts than non-profitable (1) contracts and the ratio between them is $\frac{\text{Number of non-profitable data points}}{\text{Number of profitable data points}} \approx 7$. So we are facing a class-imbalanced classification problem and traditional machine learning algorithms usually perform poorly on the data points in the minority class. Thus it will be hard for us to directly use these models to achieve the result we need. Our solution here is the multi-layer algorithm, which divide the labeling process into several different steps and we will describe the algorithm in detail later in the Section 5.2. We will present results from numerical experiments and show the effectiveness of our algorithm. We will see that our algorithm achieves great improvement over the traditional models; especially the algorithm gives pretty satisfying result for more than 85 percents of the data points in the test set

We further dig deeper into the class-imbalanced classification problems and define a second problem on the client level. In this case, we aim to label clients as historically profitable or unprofitable based on the ratio of historical loss and future premium. Based on a key observation on the classification result of traditional classifiers, we incorporate an unsupervised learning algorithm into the classical classification models to improve the performance over minority class. Numerical experiments are conducted to show the effectiveness of our finding. We will see that the hybrid model works surprisingly well and the improvements over the minority class for Support Vector Machines (SVM) is over 50 percent in terms of recall percentage. Analysis over the improvements will be provided later in the section.

The following materials of the chapter will be structured into three sections. In Section 5.2 we discuss our solution to the first problem. In Section 5.3 we will talk about the second problem and our solution to it. Finally in the last section, we will conclude the chapter and summarize our findings to the class-imbalanced classification problem.

5.2 The Multi-layer Algorithm

We call our algorithm "multi-layer" because it is layer-structured; specifically, the algorithm consists of several steps and we will label a certain portion of data points in each step and thus divide the whole dataset into multiple layers or levels. Each level or step aims to deal with data points with similar features; at least similar in the sense of classification. Different models and techniques will be applied in different layer or steps and examples include adaptive boosting and SMOTE rebalanced method.

In this section, we will first talk about the imbalanced nature of our dataset to show that the profitability prediction will always be a class-imbalanced classification problem. In Section 5.2.2 we will introduce the most important techniques include in our algorithm, adaptive boosting and the famous oversampling technique, Synthetic Minority Oversampling Technique (SMOTE), proposed by Chawla et al. (2002). We will show how these techniques would help to alleviate the problems brought by the imbalanced nature of our dataset. Later in Section 5.2.3 we will give a complete view of our algorithm and describe the whole process in detail. We conduct numerical experiments and show the effectiveness of our algorithm by confusion matrices, recall percentages and precisions.

5.2.1 Imbalanced Nature of Dataset

Based on their experience, our project collaborators provide us with the criterion that determines the profitability. We are told that if the ratio of loss and premium are less or equal to 0.35 then the contract would be profitable for them. We then label all of the data points according to the threshold; for detail label construction, please refer to Equation 3.1. As mention in Chapter 3, the corresponding target variable is called "profit label." In the previous section, we mention that we are aiming to label contracts instead of clients. This is because contract is the most basic unit in the industry and the risk control should be done over the contracts. Here a basic difference between this part of research and the reference system we build in Chapter 4 arises. The reference system is built upon the client level and a risk index is assigned to each client; however, the profitability prediction is conducted on the contract level and a profit label will be assigned to each contract.

We have in total of 4081 contracts available; 3556 of them are labeled as profitable (0) and the rest 525 are labeled as non-profitable (1). Figure 5-1 clearly shows the imbalance of 87% profitable and 13% non-profitable. We can understand the imbalanced nature intuitively; a healthy and well-operated company should be able to generate profits and thus should have more profitable contracts than non-profitable ones. Our project collaborator, as a giant in the Business to Business (B2B) insurance industry, would have even more profitable contracts than a mid-size company does.

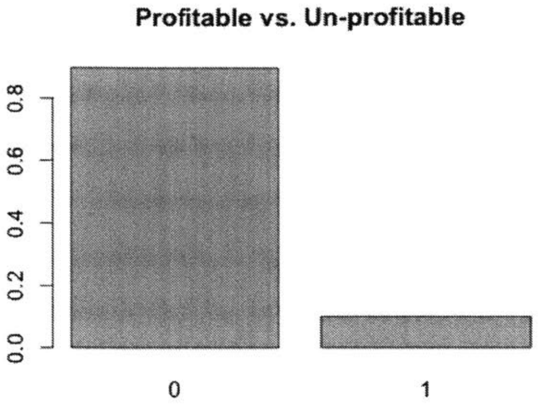


Figure 5-1: Profitable Contract vs Non-profitable Contracts

More importantly, as we mentioned in Chapter 3, we fill a certain portion of data points with zero loss when the loss information is not available. Our hypothesis is that any loss incurred by the clients would be recorded as claims. Thus clients with no loss information means that there is no claim reported and thus no loss incurred. Although this is a valid hypothesis logically, we have to admit that there are a number of contracts without loss information and thus labeled as profitable contracts. The two reasons explain the imbalanced nature of our dataset, which directly leads to our class-imbalanced problem.

5.2.2 Key Methods

As we have stated previously, adaptive boosting and Synthetic Minority Oversampling Technique (SMOTE) are the two key methods included in our algorithm. In this section, we will briefly go through these two methods and explain the way they work. In particular, we would like to show the effectiveness of the methods and explain how these advantages can be helpful for our problem; however, at the same time, we would also show their inferiority, which further explains the necessity of the multi-layer structure of our algorithm. For detailed descriptions of the actual algorithm flow and pseudo codes, please refer to Appendix A.

Adaptive Boosting

Boosting, as explained by Schapire (2013), is an approach to machine learning based on the idea of creating a highly accurate prediction rule by combining many relatively weak and inaccurate rules. The weak and maybe naive rules are often called the weaker learners and boosting aims to combine all these weak learners into a stronger and much more accurate one. In classification problems, a weak learner would be a naive classifier and boosting algorithms aim to give us a stronger predictor based on the naive ones, even weak classifiers perform as bad as random selections.

There have been a lot of boosting algorithms and the one we use is adaptive boosting, which is often referred to "AdaBoost". First introduced in 1995 by Freund and Schapire (1995), adaptive boosting is the first practical boosting algorithm and it remains one of the most widely used technique in many different fields. Bishop (2006) provides a comprehensive description for the algorithm. For binary classification problems such as the profitability prediction, AdaBoost first assigns a weight to each of the data points in the training set and train the weak classifier with the weighted dataset, where weights will be put into the loss functions. The weak classifier, also called as the base classifier, is predetermined by us and the initial weights are set to be equal to $\frac{1}{\text{Number of training data points}}$. In our case, we choose decision trees as the base learners. We then compare the predicted label with the actual label and adjust the weights according to the comparison; at the same time, another weight system will be created for the base classifiers. Accordingly, data points with wrong predicted labels will be given more weights than those with the right predicted labels; classifiers with higher prediction accuracy will be given more weights as well. Then the algorithm will train a new classifier with the newly-weighted dataset. The process goes iteratively until we reach the desired number of base classifiers. Finally, the output will be determined by the weighted sum of the results from all the classifiers. Figure 5-2 summarizes the whole process more clearly with the flow chart.

For our goal of predicting contract profitability, we are facing a series of problems that arise from the imbalanced nature of our dataset. As stated by Longadge and

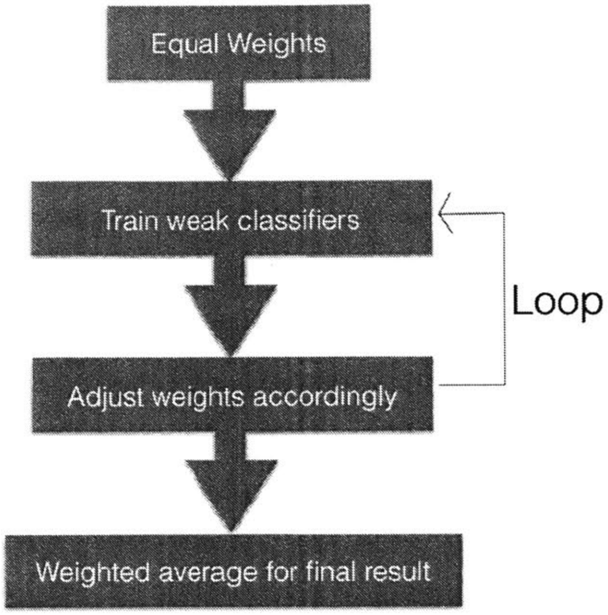


Figure 5-2: Flow Chart: Adaptive Boosting

Dongre (2013), imbalance datasets are hard to predict and most of the traditional classification algorithms would tend to ignore members in the minority class and tag everything with the same label; in our case, traditional methods would predict every contract in the test set as profitable and prediction like this is meaningless for us. Fortunately, adaptive boosting can alleviate the effect of imbalance to some extend and it has two advantages that can help with our prediction. First, AdaBoost requires no prior knowledge about the weak learner and we would not need to insist on finding an accurate labeling method for all data points. Thus we can use traditional algorithms as the weak learners and in our case, we use decision trees. Additionally, Freund et al. (1999) point out a nice property of AdaBoost: its ability to identify outliers, i.e., examples that are either mislabeled in the training data, or which are inherently ambiguous and hard to categorize. Thus data points in minority groups that are ignored by the weak learners would get higher weights and thus receive more attention. This is exactly what we want to achieve; to get those non-profitable contracts more exposed. Adaptive boosting is superior in other ways as well such as very few parameters to tune but those we just mention above are the two most important ones in our scenario.

Although adaptive boosting has advantages which fit it well into our problem, at the same time we have to admit that prediction based solely on AdaBoost is not satisfying enough. As noted by Schapire (2013), AdaBoost is very susceptible to noise, even with regularization. However, as we have stated earlier, we fill the missing label as profitable and thus increasing the number of profitable accounts. So we are actually working with a noisy dataset. The conflict generates a big problem here and thus we would not be able to use AdaBoost directly. In view of the sensitivity to noise, Friedman et al. (2000) propose gentle AdaBoost. In fact Gentle AdaBoost is a variant version of AdaBoost and the algorithm does not differ much. Friedman et al. (2000) show the difference that Gentle AdaBoost uses Newton stepping instead of exact optimization in each iteration, updates the weights more smoothly and thus puts less emphasis on outliers. The revision mitigates the effect of noisy data and helps achieve a better trade off between recall and precision. Indeed our numerical experiment shows that even with Gentle AdaBoost, the result is not as good as expected and this is an important reason for the design of a multi-layer algorithm, where gentle adaptive boosting is used partially only in certain layer.

Synthetic Minority Oversampling Technique

Scientists usually approach imbalance classification problems from two different perspectives, either from algorithm point of view or from a data perspective. Specifically, researchers would either choose to design a special algorithm against the special data structure or to operate on the original dataset to reduce the imbalance feature to a certain extend. Examples of revision on algorithms could be to penalize more on the ignorance of minority class and this is similar to what AdaBoost does as we explained in the previous section. Data operation aims to use sampling techniques to re-balance the dataset; in other words, to raise the ratio between the two classes to a value closer to 0.5. In our case, we use one of the most famous oversampling technique, the Synthetic Minority Oversampling Technique (SMOTE).

Proposed by Chawla et al. (2002), SMOTE has been applied to problems in many different areas and results have been quiet promising for many of the cases. Although only oversampling is mentioned in the name of SMOTE, the algorithm indeed includes both under-sampling and oversampling. Essentially, the method aims to help classifiers to identify decision regions for the minority class in the feature space by increasing the number of minority class members. Traditional oversampling with replacement usually operates in the data space but the oversampling technique proposed by Chawla et al. (2002) is operated in the feature space. Extra elements are added based on the positions of existing minority elements in the feature space. The algorithm would first pick a minority element at random and choose its k nearest neighbors; depending on the oversampling percentage, n data points would be generated on the line segments linking n of the k chosen neighbors and the minority sample. Specifically, if we set k to be 5 and want 200% oversampling, SMOTE would first pick a data point in the minority class and then find its 5 nearest neighbors in the feature space and then randomly choose two of the neighbors; then it will draw two lines between these two neighbors and the original minority sample. Two extra points with the minority labels would be added to the feature space locating randomly on these two lines. In this way, SMOTE forces the decision region of the minority class to become more general and more detectable. Classifiers would then define the decision region to be larger and less specific, which would then help to increase the chance of detecting more minority elements.

As we have mentioned, current SMOTE includes under-sampling to its algorithm as well. Under-sampling is operated in a straight forward way, simply remove a certain number of elements in the majority class at random. This of course would help to decrease the imbalance ratio and at the same time would help classifiers to improve its recall for the minority class. Under-sampling, although not as well-designed as the oversampling technique discussed above, is very suitable for our problem since we have a certain number of artificial majority members. By eliminating some elements in the majority class, we would also alleviate the effect of possible bad filled-in problems. It is worth noting that SMOTE has been developed overtime and many other versions of it has been release to make it fit to broader problems, such as problems with both continuous and nominal data points (SMOTE-NC).

Indeed the way we use SMOTE is quiet different from what most people do. Instead of trying to detect more elements in the minority class, we use it to label less elements in the majority class but with higher precisions. Later in the Section 5.2.3 we would show how exactly we use SMOTE in our multi-layer algorithm and we will show the effect by confusion matrices.

5.2.3 Algorithm and Computational Results

In this section, we will give a detailed description of our multi-layer algorithm and the computational results from our numerical experiments. As we mention previously, our algorithm consists of four different steps and in each step different methods will be used to label a certain portion of data points. Methods we take into our algorithm include gentle adaptive boosting, decision tree, Synthetic Minority Oversampling (SMOTE), and Random Forest. Numerical experiments are conducted based on the imbalance dataset we discuss earlier and training and test sets are divided with the ratio 3 to 1. Thus 3062 data points are included in the training set and 1019 in the test set. All experiments are conducted in the statistical computing environment R and important packages we use include "ada" for gentle adaptive boosting, "randomForest" for Random Forest classification, "e1071" for support vector machines and "DMwR" for SMOTE. For detail information, please refer to Culp et al. (2006), Liaw and Wiener (2002),Meyer et al. (2014), Friedman (2002), Torgo (2010). We've experimented hundreds of times and the result shown in this section is the average performance based on our experience.

Before heading to our algorithm, I would like to show the poor performance of traditional classification models over our dataset as a reference. Table 5.1 and Table 5.2 below provide two confusion matrices resulting from Random Forest and Support Vector Machines. As we can see, the performance of Support Vector Machines is extremely bad with zero non-profitable contract captured. Although Random Forest does a relatively better job, the overall performance over the minority class is still not acceptable with recall lower than 35 percent and precision low to 60 percent. Now we can see that non-profitable contracts in our problem are really hard to capture and each one of the capturing might cause a huge sacrifice of mistakes.

Table 5.1: Confusion Matrix from Support Vector Machines

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	919	0
Actually Non-profitable	100	0

Table 5.2: Confusion Matrix from Random Forest

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	900	19
Actually Non-profitable	68	32

In view of the hardness of prediction, we design our multi-layer algorithm with the goal to capture more non-profitable contracts with acceptable cost. We would introduce our algorithm step by step and begin with the first one. In the first step, we use decision tree together with gentle adaptive boosting as the classifier and apply the model to the whole dataset. However, we will only trust a portion of the predicted labels and all other labels would be removed. Specifically, we will take the predicted labels if they are either non-profitable or profitable with extremely high probability. Data points not within the range will continue to the second step of the algorithm. The following conditional statement shows the rule of trust or removal more clearly.

$$\text{Rule of trust or removal} = \begin{cases} \text{Trust,} & \text{"Non-profitable(1)"} \\ \text{Trust,} & \text{"Profitable"(0) with probability larger than 99\%} \\ \text{Remove,} & \text{otherwise} \end{cases} \tag{5.1}$$

The selection rule is indeed constructed based on observations during our numerical experiments and thus we would like to justify the rule with results from numerical experiments. Table 5.3 shows the confusion matrix for the original model. As we would expect from the results of Random Forests and Support Vector Machines, decision tree with gentle adaptive boosting tends to label contracts as profitable. Indeed, the result is not as bad as it could be; nearly every profitable accounts and 30 percent of the non-profitable accounts are capture. The problem occurs to the false negatives located in the lower left corner of the matrix and these are the non-profitable accounts that are mislabeled as profitable.

Table 5.3: Confusion Matrix from first step model

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	911	8
Actually Non-profitable	68	32

We notice that the precision of Non-profitable contracts remains good at around 80 percents and thus we decide to trust these predictions. Another important observations here is that when the model is very confident, its performance is indeed very satisfying. In other words, when the associated probability is high, the predicted labels would be very accurate and they are worth trusting. Thus we decide to take these two types of labels and reject all the others. After the selection process, the confusion matrix is shown below in Table 5.4. As we can see, the results for these 627 contracts are very promising.

Table 5.4: Confusion Matrix after selection

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	585	8
Actually Non-profitable	2	32

Together with selection process, the model achieves high values for both recall and precision; more importantly, the performance over the minority class is satisfying as well with precision over 94 percent and recall of 80 percents. This means that we can give a very promising result for more than 60 percent of the contracts. For undertakers, the improvement means much more than what the numbers can show; it means that for more than 60 percent of the time, they can sign the contract without much concern over the risk of loss.

Although we are happy with the result for the contracts labeled in the first step, we still need to worry about the left 392 contracts since they still count for a considerable amount of around 40 percent. Thus we continue to the second step and design a different model accordingly. In the second step, we first apply Synthetic Minority Oversampling Technique (SMOTE) to our training dataset and then use the new training set to train a new decision tree with gentle adaptive boosting. Here we use a 200% oversampling and thus raise the number of elements in the minority class to twice as many as before; at the same time, we undersample the majority class by only selecting only a portion of them into the training set, making the ratio of majority to minority class to be 1 to 10. In this way, the original majority class becomes the minority class. The performance over the remaining test set is shown below in Table 5.5.

Table 5.5: Confusion Matrix for Second Step Model

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	31	295
Actually Non-profitable	2	64

The result is indeed not surprising. SMOTE help us rebalance the dataset and now profitable contracts are the minority members and thus the model would tend to predict everything as non-profitable. In our case, we capture nearly every non-profitable contract but pay the price of losing 295 profitable contracts, which are nearly five times as many as what we achieve. This is definitely not acceptable. Fortunately, we still achieve something similar to the first step here, the 33 predicted to be profitable contracts. If we take these labels and ignore all the non-profitable data, we would not need to pay anything. Thus we decide to use SMOTE in a non-traditional way; instead of using SMOTE to capture minority labels, we use it to grasp majority class members but with extremely high precision. Since we are using the same methods as the previous step, we would not get many data labeled in this step. However, the accuracy of 31/33 ≈ 94% still makes us happy. To summarize the second step, we use the following rule in 5.2 to select label.

$$\text{Rule of trust or removal} = \begin{cases} \text{Trust}, & \text{"Profitable(0)"} \\ \text{Remove}, & \text{otherwise} \end{cases} \tag{5.2}$$

The resulting confusion matrix is shown in Table 5.6.

Table 5.6: Confusion Matrix for Second Step after Selection

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	31	0
Actually Non-profitable	2	0

With SMOTE we further construct another training set, which does not go as extreme as the one in the second step. We still oversample 200% but the undersampling coefficient is set to select more members from the majority class and this time 5 times as many as last time. Now the training data is no longer imbalanced and the model should not have a strong preference for any of the class. With this training set, we will proceed to the third step. In the beginning of this section, we show the performance of two traditional models over our dataset and we observe that Random Forest works much better than the kernel-based method. Thus we decide to take it into our algorithm as the main classifier for the third step. We train a Random Forest classifier with the rebalanced training set and apply it to the remainders in the test set. Result is again shown by a confusion matrix as in Table 5.7.

Table 5.7: Confusion Matrix for Third Step

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	165	130
Actually Non-profitable	19	45

The result is as what we expect it to be; the predicted classes have equal number of members. Still the problem here is that we are paying too much for capturing the non-profitable contracts. Thus we follow the same strategy as last step, to choose only the profitable label and leave the rest to the last step. So the result in this step would be similar to last step and it is shown in Table 5.8. Here we are detecting 165 with accuracy around 90 percent, which is still quiet satisfying.

Table 5.8: Confusion Matrix for Third Step after Selection

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	165	0
Actually Non-profitable	19	0

Till now, we have achieved good result for 83 percent of our test set. The 17 percent leftovers are indeed really hard to label with high accuracy. Thus we design a flexible model for our client to adjust according to their actual need. We use two training sets to train two decision trees with gentle adaptive boosting, calculate a weighted probability for each of the leftovers, and then label them with the weighted probability. Insurers can adjust the weights according to their actual need. If they want to detect more non-profitable contracts and are willing to sacrifice some profitable contracts they can set the weight of the rebalanced model higher; on the other hand, if they do not want to lose their potential customers and would like to undertake more risk, they can set the weight for the imbalance model higher. In Table 5.9 we show the result for the weights 0.5, 0.5.

Table 5.9: Confusion Matrix for Leftovers

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	60	70
Actually Non-profitable	18	23

We have to admit that result in this step for the 17 percent leftover is indeed bad. However, if we compare with the performance of the traditional classifiers on the whole dataset, the performance of our method on the leftovers is not as bad and still better in certain metrics. Moreover, the missing data imputation by artificially filling in zeros would probably affect the correlation of predictors to target values as well.

5.2.4 Summary

We have now introduced our multi-layer algorithm in detail and we have shown its effectiveness by applying it in numerical experiments. We achieve great results for around 83 percents of the contracts but does not have great solution for the leftovers. A huge advantage of our multi-layer algorithm is that we can divide contracts into different parts and predict with high accuracy for a large portion of them, which we can recognize with our algorithm. Thus when a new contract comes to the insurer, our algorithm would tell the insurer which step the contract would be labeled. For 83 percent of the time, a contract would be put into the first three steps and in these cases, the insurer would know that the contract can be signed without worrying too much. However, if the contract is labeled in the last step as leftovers, the insurer would need to pay more attention to the reports and maybe require more document to be submitted for further investigation. The process of our algorithm can be shown more clearly with the flow chart in Figure 5-3.

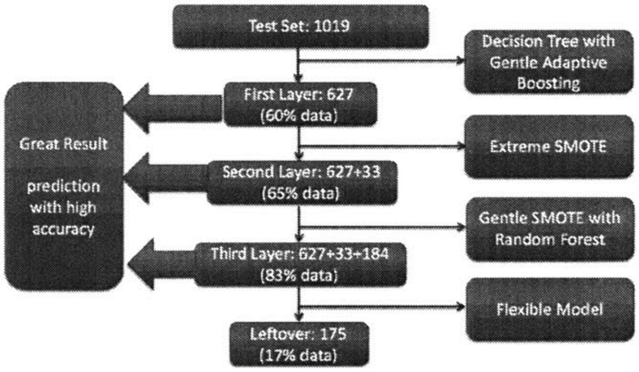

Test Set: 1019
Decision Tree with Gentle Adaptive Boosting
First Layer: 627 (60% data)
Great Result
prediction with high accuracy
Extreme SMOTE
Second Layer: 627+33 (65% data)
Gentle SMOTE with Random Forest
Third Layer: 627+33+184 (83% data)
Flexible Model
Leftover: 175 (17% data)

Figure 5-3: Flow Chart for Multi-layer Algorithm

5.3 Another Imbalance Problem

As we have stated in Section 5.1, this second problem is defined on the client level. In this case, we want to label each of the renewal accounts as historically profitable or non-profitable. Based on the historical loss incurred, which we already know, we want to tell the insurers if their renewed client would be profitable or non-profitable in the new time period as premium changes. Although this prediction might not be as significantly meaningful as the one we discuss previously, it still gives a reference to the insurers for their decision making. The problem turns out to be another class imbalance classification problem and we come up with a new algorithm which would be shown in Section 5.3.2 together with results from numerical experiments. Since we already know the amount of loss happened historically, we know half of the information contained in the target variable and thus the problem should be easier than the actual profitability prediction. Indeed, some of the existing algorithms like Random Forest does well on the classifications; however, algorithms like Support Vector Machines do not perform as well. The algorithm we propose here is to help improve the performance of classical SVM.

5.3.1 Problem

As we have stated previously, the prediction is based on historical loss and future premium. The target variable is defined according to the rule listed below. Based on our prior experience with the previous profitability prediction, we expect there would be more profitable labels than non-profitable ones. Indeed the fact coincide with our expectation. Figure 5-4 shows the distribution over the number of elements in the two groups.

$$\text{New Target Variable} = \begin{cases} 1, & \text{if } \frac{\text{Average loss incurred historically}}{\text{Premium of future year}} > 0.35. \\ 0, & \text{otherwise.} \end{cases} \tag{5.3}$$

As we mention in Chapter 4, there are 1714 data points available for client level analysis and among all these accounts, 329 of them are labeled as historically non-profitable. Figure 5-4 shows the difference in a more clear and intuitive way and we can see that the imbalance here is not as much as in the previous one. We have nearly 20 percent of the data points lying in the minority group compared to the previous problem with only 10 percents in the minority class. However, as we will show soon, it is imbalanced enough to confuse classical Support Vector Machines.

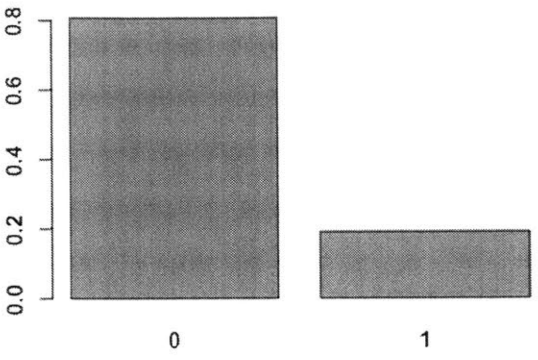


Figure 5-4: Profitable Accounts vs Non-profitable Accounts (Historically)

5.3.2 Algorithm and Computational Result

In this section, we propose a new algorithm which incorporates clustering techniques into SVM to help improve classification performances, especially on the minority class. Before heading to our algorithm, we would like to first show the performance of traditional Support Vector Machines on our problem, provide some preliminary analysis, and point out some key observations. Training and test sets are divided according to the ratio 2 to 1 and thus there would be 1149 accounts in the training set and the rest 545 are left in the test set.

We first apply Support Vector Machines directly to the dataset and the result is shown in the confusion matrix in Table 5.10. We try difference kernels such as linear, polynomial and radial basis functions and the best results come from the radial kernel. Here we only show the result from SVM with RBF. We can see that the performance this time is not as bad as in the previous problem; however, the recall percentage over the minority class is still very bad, with only 40 percent non-profitable clients being captured.

Although the result is not satisfying, it still provides some useful insights. The key observation over the result is that loss in history and historical loss frequency are the two most important indicators in this problem. Indeed, all other variables are

Table 5.10: Confusion Matrix for SVM

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	447	10
Actually Non-profitable	65	43

much less important than they are. Accordingly, we come up with the idea to first cluster our data sets by these two factors and then build two separate SVM for each clusters. Thus we first apply k-means clustering to the dataset based on historical loss percentage and historical loss frequency. The clustering results are shown in Figure 5-5 and each color represents a different cluster.

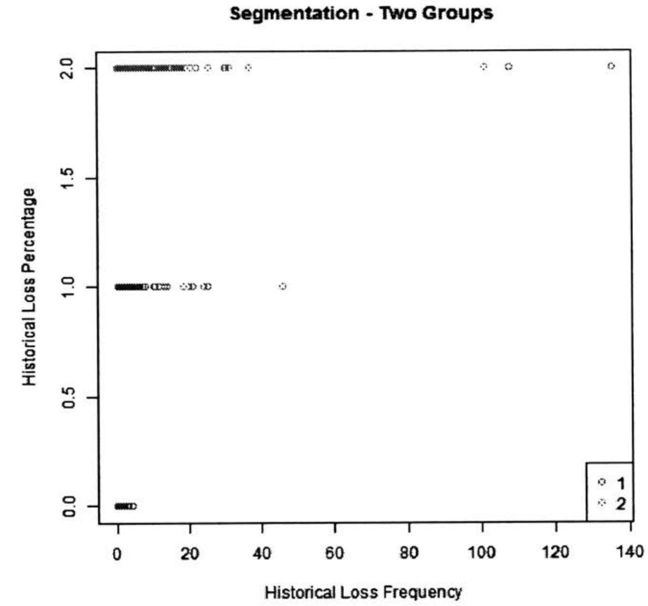
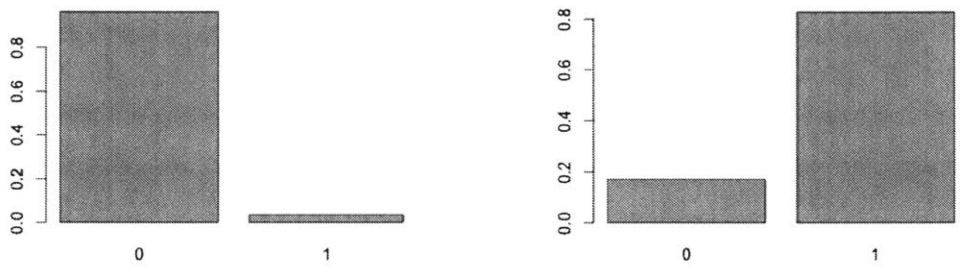


Figure 5-5: Clustering Results

We can see that the two clusters differs a lot in terms of historical loss frequency and loss percentage; the cluster lying in the lower left corner has relatively small quantities and the other with larger values has data points spread around in higher areas. With the clustering process, we expect to divide clients into two different risk groups and our expectation is that clients within each groups would share the same loss properties. Fortunately, the result is in accordance with our expectation and we are happy to see that the ratio between profitable and non-profitable labels within each cluster differs dramatically. Figure 5-6 shows the label distributions for the two clusters of our test set. Here we observe two totally different skewed distributions: profitable accounts dominant the first cluster which consists of data with low historical loss percentage and low loss frequency while non-profitable account concentrate more in the second cluster. In this way, we divide the original imbalance problem into two sub-problems, which are again imbalanced; however, in the second cluster non-profitable accounts become the majority class. Based on our experience, we will expect classical Support Vector Machines (SVM) to ignore elements in the minority class for both clusters and thus non-profitable clients in the second cluster would very likely to be detected.



(a) Label Distribution: First cluster

(b) Label Distribution: Second cluster

Figure 5-6: Comparison: Label Distribution in test set

The training set is divided in the same way and we train two SVM models with the two training sets respectively. Models are then applied to the two test sets and the resulting confusion matrices are shown below in Table 5.11 and Table 5.12.

Table 5.11: Confusion Matrix for SVM: First Cluster

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	438	0
Actually Non-profitable	3	13

The result is indeed not surprising. Traditional algorithms would always tend to ignore members in the minority class and thus non-profitable accounts in the first

Table 5.12: Confusion Matrix for SVM: Second Cluster

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	3	16
Actually Non-profitable	1	91

cluster and profitable accounts in the second cluster are ignored. We can combine them to get a overall result for the whole test set and we see an huge improvement of recall percentage in the minority group, from 39 percent to 95 percent and the precision goes up from 81 percent to 85 percent simultaneously. Overall, our proposed method makes 55 (around ten percent of the whole test set) less errors as well.

Table 5.13: Confusion Matrix for SVM with Clustering

	Predicted as Profitable	Predicted as Non-profitable
Actually Profitable	441	16
Actually Non-profitable	4	104

Our proposed method in this section is simple in structure and easy to operate. Despite the simple idea, the effect is indeed satisfying. In fact many machine learning problems are structured as this and we might be able to achieve great improvements over traditional models by easy operations. In terms of structure, the proposed model is equivalent to build Support Vector Machines classifiers upon a decision tree with one split, which is ofter referred to as the decision stump. Thus we can also view it as an extension of the traditional decision tree model with SVM as the leaf classifiers. This idea has been floating around for sometime and we are not the first to do it. The only difference made here is to use k-means clustering to replace the decision stump and we believe the unsupervised learning algorithm should lead to better splitting effect.

5.4 Conclusion

In this chapter, we tackle the problem of profitability prediction for insurers and at the same time discuss some of our thoughts about binary class-imbalanced classification problems. We design a multi-layer algorithm for our prediction task and it gives high prediction accuracy for more than 83 percent of the data points. Later in the section, we introduce a method to improve classification accuracy of SVM by incorporating k-means into the classification procedure. The improvement is impressive and the method itself is simple and easy to operate.

Imbalance classification problems are very common in a variety of industries and detecting the minority members is always the hardest part in such problems. The multi-layer algorithm is designed based on the performance of traditional classifiers on our dataset and we further include adaptive boosting and Synthetic Minority Oversampling Technique into the algorithm to help label with higher precisions. Notice that the multi-layer algorithm is indeed built upon the idea of model combination, which suggests to use different models for the task and take advantages of them to get a better result. Our algorithm here thus includes several classifiers and it take the advantages of traditional classifiers, ensemble learning methods and sampling algorithms. As shown by the confusion matrices, our multi-layer algorithm achieves different accuracy in between layers; however, for the most part, it gives satisfying results. Although the algorithm still does not resolve the whole problem completely, it gives a great solution to the most part of it. This would potentially help greatly for the decision making process in insurance companies and it would lead to a huge amount of profit generated by helping avoid many loss.

Traditional classifiers such as Support Vector Machines (SVM) usually do not perform well on the imbalance datasets but sometimes it would be easy to solve the problems incurred by traditional models. In our case, classic SVM performs bad for the designed problem; however, by simply dividing the dataset into two segments, we improve the performance by a huge amount. We show how clustering algorithm such as k-means helps to improve the recall percentage over the minority group by more than 50 percent while making more accurate predictions. Although our problem is easy and well-designed to fit our hybrid model, we still believe that it can serve as a proof of concept. Techniques such as clustering could potentially help greatly to classification problems, not restricted to imbalance classification. The idea is to try hybrid modeling which incorporates several models into an integrated model for better performance and the resulting structure might be multi-layer stack on or it could be other types.

Chapter 6

Conclusions and Future Work

This thesis was initiated by an urgent business need from insurance companies. We address the issue of client-loss pattern recognition and profitability prediction for insurers. We approach these two problems with machine learning techniques; specifically, we tackle the pattern recognition problem by building a risk index to represent the potential risk level for all clients and treat the second problem of profitability prediction as an imbalance classification task. The supervised learning task is then solved by our proposed multi-layer algorithm. We further dig deeper into the imbalance classification problems and we believe hybrid models could provide better solutions to such problems. We then incorporate unsupervised learning techniques to the traditional Support Vector Machine to prove the idea.

Based on the structure and distribution of our dataset, we choose logistic regression as our main model for the pattern recognition problem. Originally, it is structured as a multi-class classification problem, which is nearly always much harder than normal binary problems and finding a solution would require a lot more effort. Thus we approach the problem in a different way. Instead of directly taking the labels from logistic regression, we use the intermediate result, the conditional probability of each data point being in each class, and generate a risk index by calculating a weighted sum of the labels. The resulting index has been proven to be able to well represent the loss pattern of the clients and numerical experiments conducted in Chapter 4 provide strong evidence for our index outperforming the original risk score used by the insurance company.

The profitability prediction is structured as an imbalance classification problem, which is also harder than normal classifications. In view of the performance of traditional classifiers when applied to our dataset and the characteristics of certain sampling and boosting techniques, we design a multi-layer algorithm based on decision tree, Random Forest, adaptive boosting and Synthetic Minority Oversampling Technique. We show that our proposed algorithm works well for most of the data points and even for the 17 percent leftovers, we still achieve similar results as compared to traditional classifiers. The performance of our algorithm leads to huge potential benefits for insurers.

Imbalance classification is one of the areas that needs to be investigated further. As we have mentioned, these problems are very common in many different areas and thus solutions to such problems are always in need. Deeper investigation of the issues lead us to the findings that hybrid methods combining some of the existing techniques might lead to more accurate results than any one of them, and unsupervised learning methods such as clustering could potentially help with such problems. We then show an example of incorporating k-means into the classical Support Vector Machine to help improve recall percentage and accuracy over the minority class. The analysis here is still preliminary and the case we show can only be used as a proof of concept; however, we believe this might be a possible way to tackle the related problems. Moreover, during our literature review, we find feature construction to be extremely under-explored and thus it could be another direction for potential future researches.

As we have stated throughout this thesis, there have not been many researches in the area of insurance loss pattern recognition and profitability prediction. The risk index or reference system and the multi-layer algorithms we propose is just a beginning of such research and they are not perfect, despite their good performance in our case. The proposed system can serve as a proof of concept and future work could dig deeper into such issues and see if more accurate solutions could be found in a more direct way with more complete and structured datasets. Ideally, researchers could potentially come up with a measurement that is more directly related to financial losses such as the exact amount of loss. We believe that this is an under-explored field with great potential. As for the imbalance classification, we would like to suggest future research on such problems with our idea of hybrid methods. The idea of combining unsupervised learning and supervised learning algorithms is very interesting and exploration into this field might be extremely fruitful. Indeed, there have already been algorithms designed upon such ideas but they all are still not generalizable and focus on specific problems. Thus, future research could be done on the issue and design a more generalized algorithm built upon the combination idea. Additionally, feature selection techniques are potentially very powerful and researchers might also want to try applying these techniques more to various problems in machine learning and data driven projects.

Appendix A

Appendix

A.1 Algorithm: Adaptive Boosting

Here we show the algorithms for real adaptive boosting and gentle adaptive boosting with pseudo codes which is just another form to present the method. Detail descriptions can be found in chapter 5 and original discussions over the methods should refer to Schapire (2013) and Friedman et al. (2000). The below algorithms and pseudo codes are also based on the two literatures.

Algorithm 1 Real Adaptive Boosting

```
Start with equal weights: $w_i = 1/N, i = 1, 2, ..., N$
for m = 1, 2, ..., M do
    Estimate $p_m(x) = P_w(y = 1|x) \in [0, 1]$ using weights $w_i$ on the training data
    Set $f_m(x) \longleftarrow \frac{1}{2} \ln \frac{p_m(x)}{1-p_m(x)} \in R$
    Set $w_i \longleftarrow w_i \exp[-y_i f_m(x_i)], i1, 2, ..., N$, normalize so that $\sum_i w_i = 1$
end for
Output by calculating $sign[\sum_{m=1}^{M} f_m(x)]$
```

The algorithm of gentle Adaboost is very similar to real Adaboost and there is only one difference inside of each loop; in gentle adaptive boosting we use a weighted least square to solve for $f_m(x)$ and this where it makes the boosting gentle.

Algorithm 2 Gentle Adaptive Boosting

```
Start with equal weights: $w_i = 1/N, i = 1, 2, ..., N$
for m = 1, 2, ..., M do
    The function $f_m(x)$ is fitted by weighted least squares of $y_i$ to $x_i$ with weights $w_i$
    Update $F(x) \longleftarrow F(x) + f_m(x)$
    Set $w_i \longleftarrow w_i \exp[-y_i f_m(x_i)], i1, 2, ..., N$, normalize so that $\sum_i w_i = 1$
end for
Output by calculating $sign[F(x)]$
```

A.2 Algorithm: SMOTE

SMOTE is the abbreviation of Synthetic Minority Oversampling Technique, an oversampling method developed particularly for imbalance classification problems. We have introduced the method and provide some insightful discussion over it in chapter 5, here we provide the algorithm flow for the method. Original development and discussions over the method are described clearly in Chawla et al. (2002). The algorithm flow shown in the picture below is also based upon the paper.

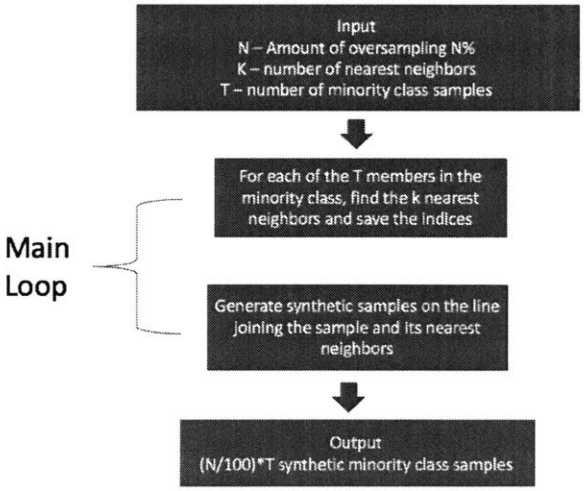

Input
N – Amount of oversampling N%
K – number of nearest neighbors
T – number of minority class samples
For each of the T members in the minority class, find the k nearest neighbors and save the indices
Main Loop
Generate synthetic samples on the line joining the sample and its nearest neighbors
Output
(N/100)*T synthetic minority class samples

Figure A-1: Algorithm: SMOTE

Bibliography

Abbott, D. (2014). *Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst*, John Wiley & Sons.

Ali, A., Shamsuddin, S. M. and Ralescu, A. L. (2015). Classification with class imbalance problem: A review, *Int. J. Advance Soft Compu. Appl* **7**(3).

Apel, H., Thieken, A. H., Merz, B. and Blöschl, G. (2004). Flood risk assessment and associated uncertainty, *Natural Hazards and Earth System Science* **4**(2): 295–308.

Baranoff, E., Brockett, P. L. and Kahane, Y. (2009). Risk management for enterprises and individuals.

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to linear optimization*, Athena Scientific, Belmont, MA, Co-published by Dynamic Ideas, LLC.

Best, A. (2008). Risk management and the rating process for insurance companies.

Beyan, C. and Fisher, R. (2015). Classifying imbalanced data sets using similarity based hierarchical decomposition, *Pattern Recognition* **48**(5): 1653–1672.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, Springer.

Chawla, N. V. (2005). Data mining for imbalanced datasets: An overview, *Data mining and knowledge discovery handbook*, Springer, pp. 853–867.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique, *Journal of artificial intelligence research* pp. 321–357.

Chawla, N. V., Japkowicz, N. and Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets, *ACM Sigkdd Explorations Newsletter* **6**(1): 1–6.

Chen, C., Liaw, A. and Breiman, L. (2004). Using random forest to learn imbalanced data, *University of California, Berkeley* .

Culp, M., Johnson, K. and Michailidis, G. (2006). ada: An r package for stochastic boosting, *Journal of Statistical Software* **17**(2): 9.

Elkan, C. (2013). Maximum likelihood, logistic regression, and stochastic gradient training.

Freund, Y., Schapire, R. and Abe, N. (1999). A short introduction to boosting, *Journal-Japanese Society For Artificial Intelligence* **14**(771-780): 1612.

Freund, Y. and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting, *European conference on computational learning theory*, Springer, pp. 23–37.

Friedman, J. H. (2002). Stochastic gradient boosting, *Computational Statistics & Data Analysis* **38**(4): 367–378.

Friedman, J., Hastie, T. and Tibshirani, R. (2001). *The elements of statistical learning*, Vol. 1, Springer series in statistics Springer, Berlin.

Friedman, J., Hastie, T., Tibshirani, R. et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), *The annals of statistics* **28**(2): 337–407.

Garg, A. and Roth, D. (2001). Understanding probabilistic classifiers.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data, *Knowledge and Data Engineering, IEEE Transactions on* **21**(9): 1263–1284.

Hérault, R. and Grandvalet, Y. (2007). Sparse probabilistic classifiers, *Proceedings of the 24th international conference on Machine learning*, ACM, pp. 337–344.

Hsu, W.-K., Huang, P.-C., Chang, C.-C., Chen, C.-W., Hung, D.-M. and Chiang, W.-L. (2011). An integrated flood risk assessment model for property insurance industry in taiwan, *Natural Hazards* **58**(3): 1295–1309.

Imam, T., Ting, K. M. and Kamruzzaman, J. (2006). z-svm: an svm for improved classification of imbalanced data, *Australasian Joint Conference on Artificial Intelligence*, Springer, pp. 264–273.

Jordan, A. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, *Advances in neural information processing systems* **14**: 841.

Kumar, A., Vembu, S., Menon, A. K. and Elkan, C. (2012). Learning and inference in probabilistic classifier chains with beam search, *Machine Learning and Knowledge Discovery in Databases*, Springer, pp. 665–680.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest, *R News* **2**(3): 18–22.
URL: *http://CRAN.R-project.org/doc/Rnews/*

Lin, K.-B., Weng, W., Lai, R. K. and Lu, P. (2014). Imbalance data classification algorithm based on svm and clustering function, *Computer Science & Education (ICCSE), 2014 9th International Conference on*, IEEE, pp. 544–548.

Linnerooth-Bayer, J., Mace, M., Verheyen, R. and Compton, K. (2003). Insurance-related actions and risk assessment in the context of the unfccc, *background paper prepared for the UNFCCC Secretariat (UNFCCC, Bonn, 2003). http://unfccc.int/meetings/workshops/other_ meetings/items/1043. php.*

Longadge, R. and Dongre, S. (2013). Class imbalance problem in data mining review, *arXiv preprint arXiv:1305.1707* .

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A. and Leisch, F. (2014). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien.* R package version 1.6-4.
URL: *http://CRAN.R-project.org/package=e1071*

Nat (2015). *State Insurance Regulation: Key Facts and Market Trends.*

Pohar, M., Blas, M. and Turk, S. (2004). Comparison of logistic regression and linear discriminant analysis: a simulation study, *Metodoloski zvezki* **1**(1): 143.

Press, S. J. and Wilson, S. (1978). Choosing between logistic regression and discriminant analysis, *Journal of the American Statistical Association* **73**(364): 699–705.

Rice, J. A. (2007). *Mathematical Statistics and Data Analysis*, third edn, Cengage Learning.

Sahare, M. and Gupta, H. (2012). A review of multi-class classification for imbalanced data, *International Journal of Advanced Computer Research* **2**(3): 160–164.

Schapire, R. E. (2013). Explaining adaboost, *Empirical inference*, Springer, pp. 37–52.

Seliya, N., Xu, Z. and Khoshgoftaar, T. M. (2008). Addressing class imbalance in non-binary classification problems, *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, Vol. 1, IEEE, pp. 460–466.

Smith, T., Hamatschek, A. and Re, S. (2007). *Getting it Right: Property Insurance Values*, Focus report, Swiss Reinsurance Company.
URL: *https://books.google.com/books?id=glgeSQAACAAJ*

Spence, R., So, E., Jenny, S., Castella, H., Ewald, M. and Booth, E. (2008). The global earthquake vulnerability estimation system (geves): an approach for earthquake risk assessment for insurance applications, *Bulletin of Earthquake Engineering* **6**(3): 463–483.

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

Sun, Y., Wong, A. K. and Kamel, M. S. (2009). Classification of imbalanced data: A review, *International Journal of Pattern Recognition and Artificial Intelligence* **23**(04): 687–719.

Tang, Y., Zhang, Y.-Q., Chawla, N. V. and Krasser, S. (2009). Svms modeling for highly imbalanced classification, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **39**(1): 281–288.

Thai-Nghe, N., Gantner, Z. and Schmidt-Thieme, L. (2011). A new evaluation measure for learning from imbalanced data, *Neural Networks (IJCNN), The 2011 International Joint Conference on*, IEEE, pp. 537–542.

Torgo, L. (2010). *Data Mining with R, learning with case studies*, Chapman and Hall/CRC.
URL: *http://www.dcc.fc.up.pt/ ltorgo/DataMiningWithR*

Tsai, C.-H. and Chen, C.-W. (2011). Development of a mechanism for typhoon-and flood-risk assessment and disaster management in the hotel industry–a case study of the hualien area, *Scandinavian Journal of Hospitality and Tourism* **11**(3): 324–341.

Visa, S. and Ralescu, A. (2005). Issues in mining imbalanced data sets-a review paper, *Proceedings of the sixteen midwest artificial intelligence and cognitive science conference*, Vol. 2005, sn, pp. 67–73.

Wang, S. and Yao, X. (2012). Multiclass imbalance problems: Analysis and potential solutions, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **42**(4): 1119–1130.

Xue, J.-H. and Hall, P. (2015). Why does rebalancing class-unbalanced data improve auc for linear discriminant analysis?, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **37**(5): 1109–1112.

Yin, L., Ge, Y., Xiao, K., Wang, X. and Quan, X. (2013). Feature selection for high-dimensional imbalanced data, *Neurocomputing* **105**: 3–11.