공학박사 학위논문

Instance-based Entropy Classifier for Imbalanced Classification Problem

데이터 불균형 해결을 위한 인스턴스 기반 엔트로피 분류기의 개발

2019 년 8 월

서울대학교 대학원

산업공학과 금융리스크공학 전공

조 풍 진

Abstract

Instance-based Entropy Classifier for Imbalanced Classification Problem

Poongjin Cho
Department of Industrial Engineering
The Graduate School
Seoul National University

Imbalanced classification, a supervised machine learning with class imbalance datasets, has been a significant problem in many areas. Due to the ignorance of minority data, a method different from the standard classification algorithm is needed. In this context, fuzzy support vector machine (FSVM) can assign the weight of each data point differently to handle the imbalanced datasets, and the studies in determining the weight have been actively conducted. In information theory, entropy possesses a descriptive power of data, and it can be employed to FSVM. To quantify the certainty of information for imbalanced classification, nearest neighbors entropy, an entropy value based on the neighbors' class, is proposed. However, the existing entropy fuzzy support vector machine (EFSVM) employs a unified neighborhood size when learning

the model, which causes misclassification. That's why this dissertation aims to develop the new instance-based classifier which better reflects neighbors' class. At first, the model of proposed instance-based entropy fuzzy support vector machine (IEFSVM) is developed based on the characteristics of nearest neighbors entropy. Given that the entropy of a fixed data point can vary according to neighborhood size, the entropy combination with several neighborhood sizes can be considered. Then, the graphical pattern of entropy combination is employed for assigning the weight with rational reasoning. Secondly, the model of IEFSVM is validated using public and real-world datasets with several benchmarks. Since the base classifier of IEFSVM is support vector machine (SVM), the benchmarks for comparison are twofold: algorithms using SVM as the base classifier and those not. Specifically, the proposed IEFSVM exhibits the statistically improved prediction performance with higher area under the receiver operating characteristic curve (AUC) than other benchmarks including EFSVM. Lastly, the model of IEFSVM is applied into Peer-to-peer (P2P) lending market to develop an investment decision model. Since the loan status of borrowers in P2P lending market is an imbalanced data, applying IEFSVM can predict fully paid loans. To enhance the profitability, a multiple regression model is also generated to detect non-default loans with high investment return. Interestingly, IEFSVM succeeds to improve the existing imbalanced classifier in terms of classification performance and even to develop an investment decision model with respect to profitability performance. In conclusion, the contribution of this dissertation involves the development

of a novel cost-sensitive classifier and the application of classifier to profitable investment decision.

Keywords: Fuzzy support vector machine, Entropy, Nearest neighbor, Imbalanced classification, P2P lending market, Investment decision, Loan status prediction

Student Number: 2013-21083

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Description

Binary classification predicts the samples into two groups. If the number of samples in a class is much larger than the other class, the data is called imbalanced data [1], and the classification problem of imbalanced datasets is considered to be a major challenge in the field of machine learning [2]. The first reason results from ignoring the data belonging to the minority group. A standard classification model tends to predict imbalanced data biased to the majority data [3], assuming that the class distribution of training data is balanced [4]. Furthermore, the standard accuracy rate, a simple measure used to evaluate the performance of a classification model, is not suitable for assessing the performance of imbalanced classification since it also scores better on the majority data. However, when dealing with imbalanced datasets, we usually have more interest in minority data, and it is hard to obtain. Therefore, the minority data should be assigned a higher priority than the majority data when

training the model [5], because if it is not well classified, it imposes significant costs. In other words, the misclassification cost for training model should be different between the two classes. Second, a study of imbalanced classification also improves handling other classification problems. For instance, in the case of a multi-class classification problem, the number of samples in each class would be different, and there are some majority and minority classes. In this case, it is better to recognize as several imbalanced binary classification problems by combining the two classes rather than perceiving it as a multi-class classification problem in terms of performance and comparability [6]. Therefore, innumerable researches have been developed on how to handle imbalanced datasets and its applications to other fields.

Among them, fuzzy support vector machine (FSVM) is known as one of the widely accepted methods of handling imbalanced data. While support vector machine (SVM) gives equal importance to all samples when determining the support vectors, FSVM can assign the weight differently for each sample [7], and the weight is called fuzzy membership. Specifically for imbalanced data, FSVM gives high importance to the minority data and low importance to the majority data. Then, there are numerous researches on how to set the value of importance for each sample when training FSVM.

To allocate the importance properly, entropy, which is known to possess a descriptive power of data and to quantify the certainty of information [8],

is introduced. The concept of entropy began in physics, but it has been established through information theory, and many studies have been carried out in conjunction with the nearest neighbor concept in the field of data mining. Especially, there has been an attempt to calculate the entropy value by identifying the class element of the nearest neighbors. For example, Chen *et al.* [9] initially present a method of quantifying uncertainty using entropy in a neighborhood system, which is called nearest neighbors entropy, and Fan *et al.* [10] propose entropy fuzzy support vector machine (EFSVM), which assigns fuzzy membership of FSVM using nearest neighbors entropy. Since nearest neighbors entropy includes information about certainty of sample's class, it can be employed to formulate the fuzzy membership of FSVM. Fan *et al.* [10] verify the classification performance of EFSVM through area under the receiver operating characteristic curve (AUC) [11] with other classification benchmarks, and EFSVM adequately handles imbalanced data.

In this dissertation, EFSVM is chosen as a base model because of its great ability to solve imbalanced problems. However, the model uses a uniform neighborhood size when evaluating nearest neighbors entropy, which can lead to misclassification [12]. In other words, the parameter of the model is the number of nearest neighbors, k. The small k value reflects information only from near sample, so the useful information may not exist and overfitting can occur, whereas the large k value can include outliers in the k nearest neighbors, and

complex distributions are difficult to reflect [13]. Thus, in order to develop a classifier robust to the change of neighborhood size, there have been attempts to combine information according to neighborhood size in many literatures [12, 14, 15, 16, 17]. In other words, instead of going through a tuning process where a uniform k value is determined, it is sensible to combine the information by various neighborhood size to formulate the fuzzy membership. Hence, this dissertation proposes a novel fuzzy membership evaluation with an appropriate combination of the nearest neighbors entropy, which is called instance-based entropy fuzzy support vector machine (IEFSVM). The fuzzy membership of IEFSVM is based on the graphical pattern of nearest neighbors entropy. To demonstrate the performance of IEFSVM, we tried to compare the public and real imbalanced datasets with several benchmark algorithms. In addition, for Peer-to-peer (P2P) lending market among real datasets, we have further investigated whether IEFSVM will have a great performance for classification as well as profitability.

P2P lending is one of the most well-known FinTech's financial technology that links individual investors with loan borrowers via online platform. Loan borrowers apply for loans to online platform, providing their personal and financial information. The online platform then pre-evaluates the borrowers' credit through their own algorithm before collecting individual investors, rejecting borrowers' loan applications with very poor credit [18]. It also eval-

uates grade of borrowers' credit and openly posts their credit grade, personal information, financial information, interest rate, and loan amount on their online platform. Then, individual investors can use the information to determine which loans to invest and how much to invest. The investment return of investors is determined by two factors, the interest rate and the amount of borrowers' repayment. The borrowers' interest rate is determined by the credit grade assigned by the online platform, and borrowers have to pay back the principal plus the interest rate. Then, the maximum value of investors' revenue is the interest rate of loans. However, borrowers may not be able to repay all of their money, and if the default is declared, the investors may not be able to recover the principal [19]. Therefore, investors should keep such risks in mind [19]. Despite the risk, P2P lending market is attracting investors because they can expect higher investment return than commercial banks and develop new revenue models in a stagnant financial market. This is because P2P lending company only needs commission for evaluating the credit grade and maintaining the online platform. Furthermore, factors such as the deregulation of financial institutions, technological improvement, and easy access via online and small investment [20] attract investors' attention. In addition, borrowers can request money at a lower interest rate than other financial institutions. Thus, these factors lead to numerous research on P2P lending market.

Generally, various researches on P2P lending market have been suggested,

and the main research topics include the loan status prediction and investment decision. First, the loan status prediction problem is a binary classification problem that predicts the loan status based on personal and financial information of borrowers. The borrowers' information includes annual income, employment length, interest rate, number of open account, revolving utilization rate, and et cetera, while the loan status consists of fully paid loans and default loans. Actually, the particular aspect of loan status is that the number of fully paid loans is much larger than that of default loans. For instance, in case of Lending Club, the largest P2P lending platform, the fully paid loans occupy 87.4%, whereas the default loans have 12.6%. This feature leads to the incorrect prediction through standard classification algorithms, and requires appropriate models for imbalanced data. In other words, the problem is to separate the default loans belonging to the minority group from the fully paid loans belonging to the majority group.

Alternatively, interest rate, one of the factors affecting investment return, is determined by the borrower's risk, which is assessed according to the own algorithm of P2P platform. For example, Lending Club evaluates the grade of borrowers from A to G. In detail, it is from A1 to A5, ..., G1 to G5. The closer to grade A, the more credible the borrowers are, and thus borrowers can borrow large amounts of money at a low interest rate. Conversely, if the grade is close to G, the borrowers' credibility is doubtful, and then the borrowers are charged

with a high interest rate and can borrow only a small amount of money. In the case of investors, the average investment return and risk of money being repaid can be considered for each grade, and of course the average return and risk vary by grade. In case of grade A, the average interest rate is 7.52%, while the average investment return is 7.12%, not much difference between the two. This is because in almost all cases are fully paid because borrowers with grade A are credible. On the other hand, for grade G, the average interest rate is 23.75%, whereas the average investment return is 11.59%, a large difference between the two. This is unreliable for borrowers with grade G, and in almost all cases, the loans are defaulted. Furthermore, the average risk is 37.59%, much higher than that of A-grade, 13.25%. If the loans close to grade G are to be fully paid, then a high investment return will be obtained. Thus, the grading system allows a portfolio composition according to risk aversion.

The loan status prediction problem in P2P lending market is analogous to credit scoring problem in fixed income markets in that imbalanced classification is required. Since the study on P2P lending market has been a limited understanding rather than the credit scoring problem, the overall review on credit scoring problem will greatly contribute to the study of P2P lending market and Marques *et al.* [21] summarized the evolution of credit scoring problem. In the case of the credit scoring problem, not only the minority class is separated from the majority class through algorithms suitable for imbalanced classifica-

tion, but profit-based models have been developed to aid individual investors for investment decisions [22, 23, 24]. Hence, this dissertation proposes an investment decision in P2P lending market, which employs IEFSVM for loan status prediction.

1.2 Research Motivation

The first motivation of this dissertation is to improve the classification performance of loan status prediction problem. Fan *et al.* [10] proposed a novel fuzzy membership evaluation using nearest neighbors entropy, which can quantify certainty of information. This model demonstrates the great performance of imbalanced classification, and EFSVM is declared as a base model of this dissertation. Despite their effectiveness of imbalanced classification, the drawback of EFSVM is the neighborhood size issue. The process of EFSVM includes the tuning of neighborhood size, which finds the training model with the lowest error sum of test set. However, such tuning leads to unified neighborhood size for all samples, and some samples may result in misclassification [12]. Even more, data with complex distribution will not be tuned well with high error sum of test set. In this case, imbalanced data cannot be widely analyzed via EFSVM. In addition, most of the existing researches [25, 26, 27, 28, 29] assigned fuzzy membership, focusing on proposing a novel function of neighborhood size. To cope with the limitation, some empirical researches of nearest neighbors into classification propose instance-weighting, without much dependence on the neighborhood size [12, 14, 15, 16, 17]. Instead of tuning the number of nearest neighbors as a fixed value, they consider the change of class elements according to neighborhood size, and combine the information to assign

fuzzy membership of each sample. In this manner, this dissertation attempts to better reflect nearest neighbors' information by proposing instance-based fuzzy membership evaluation, proposed in [30], which employs the pattern of nearest neighbors entropy. For a fixed point, nearest neighbors entropy can be computed to several values according to neighborhood size. When the entropies are called entropy pairs, the entropy pairs have a specific graphical pattern, which can lead to a novel instance-based fuzzy membership evaluation. Once the model is logically established with the pattern, the validity of proposed model is tested with various public and real-world imbalanced datasets.

The second motivation of this dissertation is to enhance the performance of investment decision model in P2P lending market. In previous researches, there have been two major studies to develop the investment decisions, and both are selected as base models of this dissertation. First, Serrano-Cinca and Gutierrez-Nieto [31] proposed a profit scoring concept using the internal rate of return (IRR). Most studies on lending markets have set the classification of loan status as a research direction, but they aimed at forecasting the IRR that implies the loan profitability. Specifically, they constructed a portfolio with loans expected to have a high IRR via regression model. This study suggests that investing in loans with high expected return using simple regression, regardless of risk, is on average profitable. Second, Guo *et al.* [32] evaluated the credit risk with an instance-based model by kernel regression. Since the

model adapted the kernel weight of each loan, they can optimize the portfolio, predicting the return and risk of each loan. Also, these researches assessed the performances of their investment decision via the investment return and Sharpe ratio [33]. Alternatively, some studies have applied a novel classifier for imbalanced data and constructed a simple portfolio. Xia *et al.* [34] proposed a classifier combining the cost-sensitive learning and extreme gradient boosting (XGBoost), and developed a portfolio allocation model with boundary constraints. In this manner, this dissertation proposes an investment decision model in P2P lending market, which employs IEFSVM for selecting loans to be fully paid and predicts investment return of chosen loans with regression model.

Based on two motivations, the contribution of this dissertation is incorporating a novel cost-sensitive loan status prediction into an investment decision particularly for P2P lending market. The practical contribution of this dissertation is proposing a concept to predict fully paid loans by IEFSVM. In addition, regression model is employed to rank loans that are expected to yield a high investment return. Constructing the loans classified as fully paid and predicted high investment return into the portfolio, our investment decision is expected to realize a high Sharpe ratio. The technical contribution is to enhance the investment decision model proposed by Serrano-Cinca and Gutierrez-Nieto [31] by selecting loans classified as fully paid using IEFSVM.

1.3 Organization of the Thesis

The rest of this dissertation is organized as follows. Chapter 2 introduces the related previous literatures for the entropy in a neighborhood system, imbalanced classification with nearest neighbors entropy, and P2P lending market. Chapter 3 focuses on the evaluation and characteristics of nearest neighbors entropy. In Chapter 4, based on the graphical pattern of nearest neighbors entropy, the description and empirical results of the proposed IEFSVM are provided with public and real-world imbalanced datasets. Specifically, since the proposed IEFSVM is a modified model of the existing EFSVM, the comparison of both models is carefully discussed. Chapter 5 demonstrates the imbalanced characteristics of P2P lending market, and the novel investment decision model using the proposed IEFSVM in P2P lending market is presented in comparison with several benchmarks. Finally, the contributions and limitations of this dissertation are provided in Chapter 6 with concluding remarks and possible future research for improvement.

Chapter 2

Literature review

2.1 Neighborhood system

The nearest neighbor concept has been employed to assign the weight for each sample since it can reflect the surrounding information. For instance, Zhu *et al.* [28] proposed a nearest neighbor chain, which sequentially links the nearest neighbors of opposite class. The chain is comparable to decision plane, and it can be used to allocate the weight for each sample. Moreover, nearest neighbors can allocate the weight of twin support vector such as weighted rough v-twin support vector machine [35] and structural twin support vector machine [36]. Furthermore, some studies employed the angle between a neighbor and the central point of neighbors to weighted one-class support vector machine [25], sample reduction [26], and boundary detection [27] for support vector machine.

The early models of instance-weighting focused on developing appropriate function of neighborhood size [25, 26, 27, 28, 29]. To cope with the drawback of using unified neighborhood size, several researches have developed a clas-

sifier without much dependence on the neighborhood size. Zhang *et al.* [16] developed dynamic local neighbhorhood, which evaluated the posterior probability of query samples for minority class by using the positive-negative border of each sample. Zhu *et al.* [15] employed the modified law of gravitation to calculate the distance of fixed radius nearest neighbors without tuning any parameters. Ertugrul and Tagluk [14] utilized the adaptive dependency region to evaluate the similarity and dependency of nearest neighbors using the distance and angle between two samples, respectively. Pan *et al.* [12] predicted the class via measuring the harmonic mean distance of nearest neighbors.

2.2 Imbalanced classification with entropy

Imbalanced classification problem has been considered as an important problem in many areas in the real-world such as biology [37, 38], ecology [39, 40], finance [41, 42], marketing [43, 44], medicine [45, 46], telecommunication [47, 48] and the web [49, 50, 51]. Among methods of approaching the imbalanced classification problem, FSVM is a well-known method, and many papers have developed the ways to determine the fuzzy membership. Specifically, several fuzzy membership evaluation methods are related to distance concept such as employing the distance from the separating hyperplane and center of the class [52], measuring the distance of samples after converting with kernel function [53], and considering the decaying function with distance [54]. Alternatively, Hwang *et al.* [55] utlized the imbalance ratio (IR), the ratio of the number of minority class samples and the majority ones, to determine the fuzzy membership, i.e. $IR = n_{min}/n_{maj}$ where n_{min} and n_{maj} are the number of minority and majority samples, respectively.

There have also been attempts to solve the imbalanced classification problem using nearest neighbors such as learning the weight with nearest neighbor density estimation [56] and employing the synthetized neighborhoods to ensemble learning [57, 58]. Advancing the quantification of information's certainty, numerous researches incorporated nearest neighbors and entropy concept. For

instance, Kaleli [59] collected the most similar neighbors via the smallest entropy difference. Zheng and Zhu [60] introduced the intuitionistic fuzzy entropy in the neighborhood system, whereas Chen *et al.* [61] proposed a novel uncertainty concept in the neighborhood system such as information quantity, information granularity, and nearest neighbors entropy. Among them, Fan *et al.* [10] utilized the nearest neighbors entropy to EFSVM. Furthermore, as an extension of EFSVM, Zhu and Wang [62] introduced an entropy-based matrix learning machine with Matrix-pattern-oriented Ho-Kashyap learning machine (MatMHKS), while Gupta *et al.* [63, 64] incorporated EFSVM and twin support vector machine.

The mechanism of new model, IEFSVM, follows the work of Cho *et al.* [30], and the validity of performance is demonstrated through several benchmarks. The first comparison is performed with SVM-based classifiers such as canonical SVM [65], SVM with undersampling (u-SVM), cost-sensitive SVM (cs-SVM), FSVM [52], and EFSVM [10]. While u-SVM is a resampling classifier with preprocessing method [56], cs-SVM, FSVM, and EFSVM belong to cost-sensitive classifiers. The second comparison is implemented with other classifiers such as cost-sensitive adaptive boosting (cs-AdaBoost) [66], cost-sensitive Random Forest (cs-RF) [67], EasyEnsemble [68], random under-sampling boosting (RUSBoost) [69], weighted extreme learning machine (w-ELM) [70], and cost-sensitive extreme gradient boosting (cs-XGBoost) [34]. Specifically, EasyEnsemble and

RUSBoost are affiliated to ensemble learning and used as benchmarks in many literatures [71, 72, 73, 74]. These two benchmarks combine random undersampling and boosting algorithms, in which EasyEnsemble selects several subsets from the majority class, whereas RUSBoost randomly discards samples from the majority class until the certain balance is achieved. Alternatively, w-ELM generalizes the single hidden layer feedforward networks (SLFNs) with an extra weight [75], and it is robust to both balanced and imbalanced data. All of these benchmarks in the second comparison employ other base classfiers such as boosted ensemble, decision tree, and neural network. Therefore, the proposed IEFSVM will be fully compared with these benchmarks. For the criterion of performance, the area under the receiver operating characteristic curve (AUC) is utilized, generally used for imbalanced classification [76].

2.3 P2P lending market

The early models of P2P lending market tried to characterize the default factors. For instance, Serrano-Cinca *et al.* [77] employed survival analysis with a hypotheses test to discover the factors, whereas Jiang *et al.* [78] derived the factors from a soft information of descriptive loan text using latent Dirichlet allocation (LDA) model.

Advancing the loan status prediction problem, it is widely accepted fact that the problem is comparable to credit scoring. Therefore, studies on the development of credit scoring will lend aid to understanding the P2P lending market [79, 80, 81, 82, 83, 84, 85, 86]. Among the studies, the application of imbalanced classification holds the foremost position. For example, Marques *et al.* [87] validated several re-sampling algorithms of class imbalance problem in credit scoring, whereas Sun *et al.* [88] incorporated the synthetic minority oversampling technique (SMOTE) and Bagging ensemble to predict imbalanced financial distress.

In the latest studies, the loan status prediction problem has been developed in the basis of the newest models such as soft information from descriptive text [78], heterogeneous ensemble [89], contrastive pessimistic likelihood estimation with gradient boosting [90], cost-sensitive version of extreme gradient boosting (XGBoost) [34], and Bayesian hyper-parameter optimization of XGBoost [91].

Alternatively, numerous models have been developed to increase the profitability. For instance, Zeng *et al.* [92] constructed a bipartite graph between individual investors and borrowers' loans, and developed an investment decision with iteration computation approach. Also, the benchmark models of Serrano-Cinca and Gutierrez-Nieto [31] and Guo *et al.* [32] also aims at enhancing the profitability.

As mentioned above, in order to validate the effectiveness of proposed model in P2P lending market, two types of benchmark are considered: classification and profitability. First, the benchmarks of imbalanced classification for loan status prediction consist of six state-of-the-art algorithms including cs-AdaBoost, cs-RF, EasyEnsemble, RUSBoost, w-ELM, and cs-XGBoost. The performance measures include AUC, precison, and predicted negative condition rate. The precision and predicted negative condition rate are extra ratios to examine whether an imbalanced classification demonstrates a decent performance in loan status prediction. Also, top decile [93] is considered by employing the top 10% of samples predicted as yielding a high investment return. In fact, the top decile is often used to measure the minority samples in churn prediction [94], however, the proposed model revises the top decile to evaluate the investment return. Second, the benchmarks of profitability for investment decision are comprised of two models including Serrano-Cinca and Gutierrez-Nieto [31] and Guo *et al.* [32]. The performance measures contain

the investment return and Sharpe ratio [33]. Note that Sharpe ratio is the average investment return per unit total risk. Thus, high Sharpe ratio indicates attractive risk-adjusted return of investment decision model. In this manner, the performances of proposed model are compared in terms of classification and profitability.

Chapter 3

Nearest neighbors entropy

3.1 Evaluation of nearest neighbors entropy

Entropy can quantify the certainty of information [8], and nearest neighbors entropy employs the nearest neighbors of each sample to represent the certainty of information as an entropy value [9]. The information possessed by the nearest neighbors entropy is a measure of which class the sample belongs to, and can be evaluated regardless of the number of classes. In this study, we confine to binary classification, and for convenience, we will call the two classes as positive and negative classes. Then, the nearest neighbors entropy in binary classification is defined as follows.

$$H_i = \begin{cases} 0 & \text{if } p_i = 0 \text{ or } q_i = 0 \\ -p_i \ln(p_i) - q_i \ln(q_i) & \text{otherwise} \end{cases} \tag{3.1}$$

$$p_i = N_i^+/k, \quad q_i = N_i^-/k \tag{3.2}$$

where p_i and q_i denote the probabilities of belonging to positive and negative class, respectively. k is the number of nearest neighbors. Note that N_i^+ and N_i^- are the number of positive and negative elements among k nearest neighbors, respectively.

By searching k nearest neighbors for each sample, we can calculate the entropy value by examining the neighbors' class and evaluating each probability. Even if the data is fixed, nearest neighbors entropy can vary depending on k value. Then, we define nearest neighbors entropy obtained using k nearest neighbors for sample i as $H_{i,k}$. To give a full detail of nearest neighbors entropy, we provide two examples of the calculating process of entropy. In Figure 3.2, for example, we search 15 nearest neighbors for sample i, and examine the class of the neighbors. The neighbors belonging to the positive class and the negative class are represented by triangles and squares, respectively, which consist of eight triangles and seven squares. According to the above equation, the entropy with 15 nearest neighbors is $H_{i,15} = -\frac{8}{15}\ln(\frac{8}{15}) - \frac{7}{15}\ln(\frac{7}{15}) = 0.6909$. In the same way, we can evaluate the nearest neighbors entropy with other k values. For example, the entropy with 13 nearest neighbors is $H_{i,13} = -\frac{8}{13}\ln(\frac{8}{13}) - \frac{5}{13}\ln(\frac{5}{13}) = 0.6663$. The entropies with 11 and 9 nearest neighbors are $H_{i,11} = -\frac{8}{11}\ln(\frac{8}{11}) - \frac{3}{11}\ln(\frac{3}{11}) = 0.5860$ and $H_{i,9} = -\frac{8}{9}\ln(\frac{8}{9}) - \frac{1}{9}\ln(\frac{1}{9}) = 0.3488$, respectively. When the neighborhood size is 1, 3, 5, 7, there is no negative element, so $H_{i,1} = H_{i,3} = H_{i,5} = H_{i,7} = 0$.

Therefore, for the same sample, the entropy value depends on how the value of k is set.

The second example is for a fixed neighborhood size as in Table 3.1. If the neighborhood size is 9, the number of positive elements among 9 nearest neighbors, N_i^+, can vary from 0 to 9. Subsequently, the probabilities of belonging to positive class, p_i, can be $\{0, \frac{1}{9}, \frac{2}{9}, \cdots, \frac{8}{9}, 1\}$, and the corresponding entropy values are identified in the third column of Table 3.1. Thus, according to the two examples, the nearest neighbors entropy is dependent on the data point and neighborhood size.

To be more specific about nearest neighbors entropy, we examine the entropy values that can be calculated. When we identify one neighbor, the neighbor can belong to positive or negative class, which are two cases. For example, if we denote one positive element and zero negative element to $(1, 0)$, there can be $(1, 0)$ and $(0, 1)$ with one neighbor as in Figure 3.3. If the neighborhood size increases to three, two additional neighbors' classes have to be determined. For instance, $(1, 0)$ can be three cases, which are $(3, 0)$, $(2, 1)$, and $(1, 2)$. However, $(1, 0)$ cannot be $(0, 3)$ since the class of the inner neighbor is fixed. Likewise, $(0, 1)$ can be also three cases, which are $(0, 3)$, $(1, 2)$, and $(2, 1)$. However, $(0, 1)$ cannot be $(3, 0)$. In this context, combinations of $\{H_{i,1}, H_{i,3}\}$ can be $2 \times 3 = 6$ cases. In the same way, we can observe $2 \times 3^7 = 4374$ cases from combinations of $\{H_{i,1}, H_{i,3}, H_{i,5}, ..., H_{i,15}\}$. We will call $\{H_{i,1}, H_{i,3}, H_{i,5}, ..., H_{i,15}\}$ as entropy

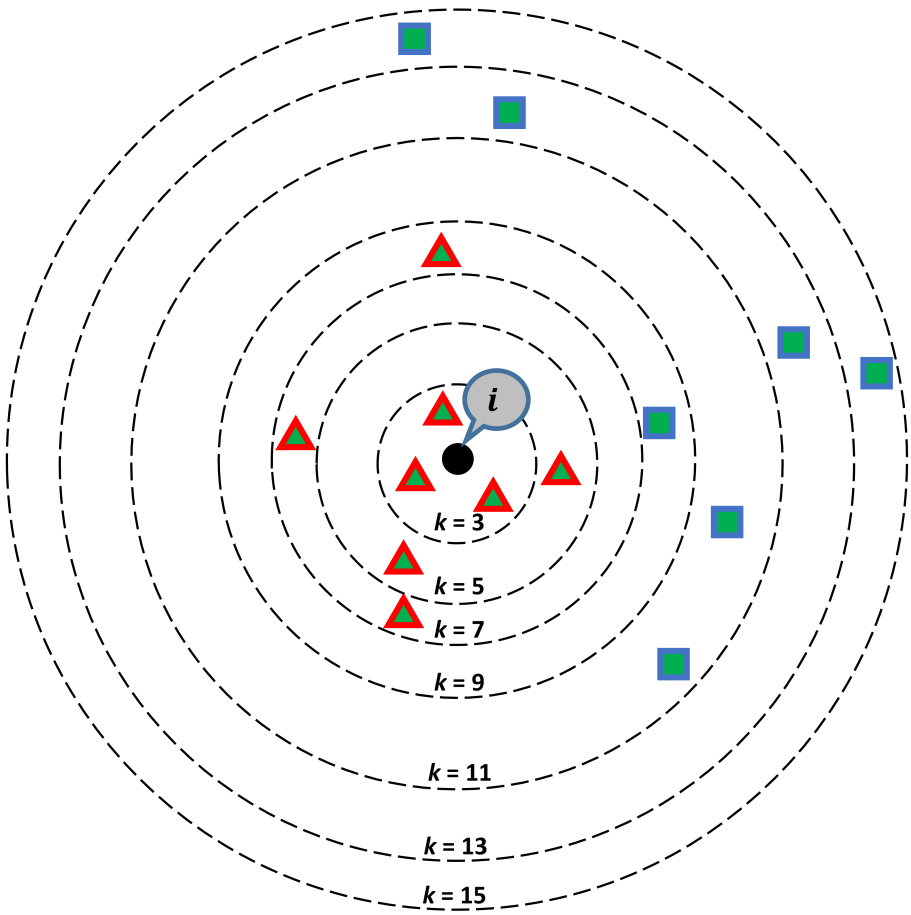

Figure 3.1: Neighborhood for a data point

Table 3.1: Example of nearest neighbors entropy for a fixed neighborhood size

N_i^+	p_i	$H_{i,9}$
0	0	0
1	$\frac{1}{9}$	$-\frac{1}{9}\ln(\frac{1}{9}) - \frac{8}{9}\ln(\frac{8}{9}) = 0.3488$
2	$\frac{2}{9}$	$-\frac{2}{9}\ln(\frac{2}{9}) - \frac{7}{9}\ln(\frac{7}{9}) = 0.5297$
3	$\frac{3}{9}$	$-\frac{3}{9}\ln(\frac{3}{9}) - \frac{6}{9}\ln(\frac{6}{9}) = 0.6365$
4	$\frac{4}{9}$	$-\frac{4}{9}\ln(\frac{4}{9}) - \frac{5}{9}\ln(\frac{5}{9}) = 0.6870$
5	$\frac{5}{9}$	$-\frac{5}{9}\ln(\frac{5}{9}) - \frac{4}{9}\ln(\frac{4}{9}) = 0.6870$
6	$\frac{6}{9}$	$-\frac{6}{9}\ln(\frac{6}{9}) - \frac{3}{9}\ln(\frac{3}{9}) = 0.6365$
7	$\frac{7}{9}$	$-\frac{7}{9}\ln(\frac{7}{9}) - \frac{2}{9}\ln(\frac{2}{9}) = 0.5297$
8	$\frac{8}{9}$	$-\frac{8}{9}\ln(\frac{8}{9}) - \frac{1}{9}\ln(\frac{1}{9}) = 0.3488$
9	1	0

pairs, and Table 3.2 demonstrates all the combinations of entropy pairs.

Table 3.2 enumerates all entropy pairs of $\{H_{i,1}, H_{i,3}, H_{i,5}, ..., H_{i,15}\}$, and first column counts all cases. Due to the clarity, we also indicate the case of Figure 3.2, placed on the second row from the bottom. The second through ninth columns indicate the entropy values according to neighborhood size, and the numbers in parentheses represent the number of elements belonging to the positive and negative class among the neighbors. As mentioned above, when

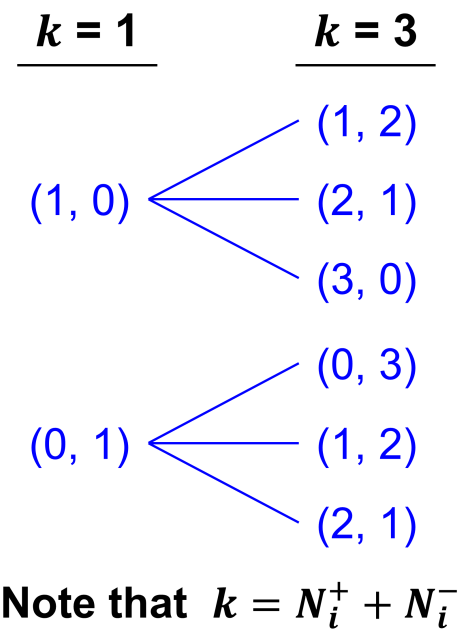

Figure 3.2: Example of (N_i^+, N_i^-) of the case for neighborhood size $\{1, 3\}$

the neighborhood size increases to next column, only two additional neighbors' classes are determined. Also, we evaluate the mean and standard deviation of entropy pairs in tenth and eleventh columns by following equation.

$$\mu_i = \sum_{k=1}^{8} H_{i,2k-1}/8, \quad \sigma_i = \sqrt{\sum_{k=1}^{8} (H_{i,2k-1} - \mu_i)^2/7} \tag{3.3}$$

Table 3.2: Enumeration of all nearest neighbors entropy (Note that: N_i^+ and N_i^- are written in parentheses)

i	$H_{i,1}$	$H_{i,3}$	$H_{i,5}$	$H_{i,7}$	$H_{i,9}$	$H_{i,11}$	$H_{i,13}$	$H_{i,15}$	μ_i	σ_i
1	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0 (0, 13)	0 (0, 15)	0	0
2	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0 (0, 13)	0.2449 (1, 14)	0.0306	0.0866
3	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0 (0, 13)	0.3927 (2, 13)	0.0491	0.1388
4	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.2712 (1, 12)	0.2449 (1, 14)	0.0645	0.1197
5	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.2712 (1, 12)	0.3927 (2, 13)	0.0830	0.1570
6	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.2712 (1, 12)	0.5004 (3, 12)	0.0964	0.1888
7	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.4293 (2, 11)	0.3927 (2, 13)	0.1027	0.1905
8	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.4293 (2, 11)	0.5004 (3, 12)	0.1162	0.2160
9	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.4293 (2, 11)	0.5799 (4, 11)	0.1262	0.2370
10	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0.3046 (1, 10)	0.2712 (1, 12)	0.2449 (1, 14)	0.1026	0.1425
11	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0.3046 (1, 10)	0.2712 (1, 12)	0.3927 (2, 13)	0.1211	0.1704
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Fig. 3.2.	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0.3046 (10, 1)	0.5402 (10, 3)	0.6365 (10, 5)	0.1852	0.2714
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
4374	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0 (11, 0)	0 (13, 0)	0 (15, 0)	0	0

3.2 Graphical pattern of nearest neighbors entropy

In section 3.1, we examine all entropy values according to neighborhood size. Then, in order to discover the graphical pattern of nearest neighbors entropy, for all 4374 samples in Table 3.2, we can draw a scatterplot (μ_i, σ_i) as in Figure 3.3. The x and y axes of the scatterplot are the mean and standard deviation of entropy pairs, respectively.

In Figure 3.3, the first (μ_i, σ_i) scatterplot is for the odd number of neighborhood sizes from 1 to 15, *i.e.* $\{1, 3, 5, ..., 15\}$. The second and third (μ_i, σ_i) scatterplots are for the natural numbers of neighborhood sizes from 1 to 15, *i.e.* $\{1, 2, 3, ..., 15\}$, and from 1 to 20, *i.e.* $\{1, 2, 3, ..., 20\}$, respectively. The more neighborhood sizes we employ, the denser the scatterplot is. Generally, all three scatterplots seem fan-shaped, and we can observe that there are no points outside a particular sector boundary. Taking into account that these scatterplots have a fan shape, we introduce a polar coordinate method. Specifically, we transform (μ_i, σ_i) into (d_i, θ_i) as in Eq.(3.4), and Figure 3.4 complements the concept of polar coordinate. From here, we set the neighborhood sizes to $\{1, 3, 5, ..., 15\}$.

$$d_i = (\mu_i^2 + \sigma_i^2)^{\frac{1}{2}}, \quad \theta_i = \tan^{-1}(\mu_i/\sigma_i) \tag{3.4}$$

Alternatively, these scatterplots appear a set of lines to the origin. Then, it is necessary to analyse the common features of points belonging to the

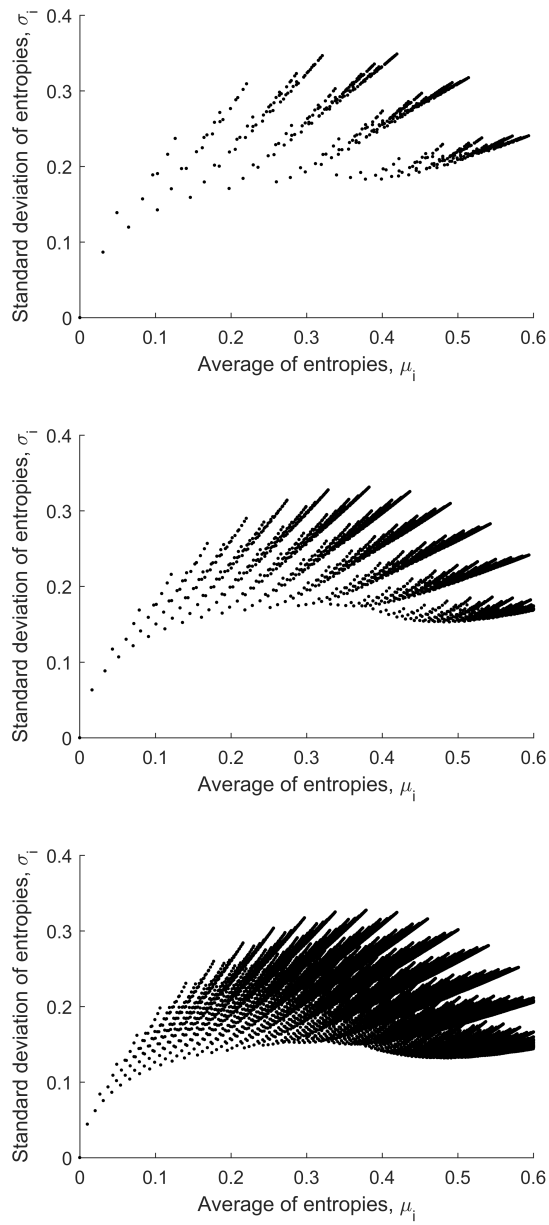

Figure 3.3: Scatterplot of mean and standard deviation of all nearest neighbors entropy with varying neighborhood sizes

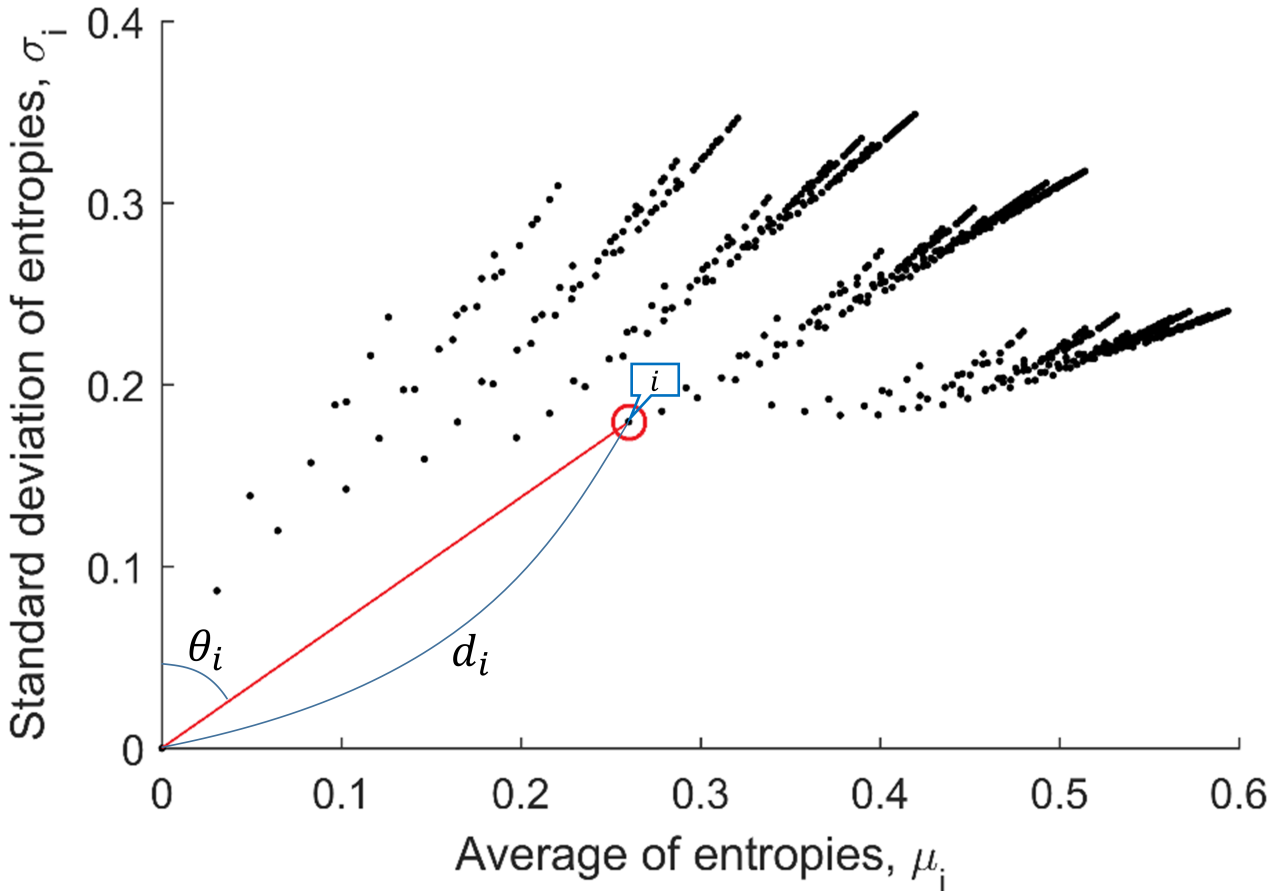

Figure 3.4: Scatterplot with polar coordinates

same line, and the differences between the lines. To be specific, we explore the number of nonzero entropy in entropy pairs. For example, the entropy pair of Figure 3.2 is $\{0, 0, 0, 0, 0.3488, 0.5860, 0.6663, 0.6909\}$, then the number of nonzero entropy in the entropy pair is four (i.e. $0.3488, 0.5860, 0.6663$, and 0.6909). In the same way, we can enumerate entropy pairs with one and two nonzero entropies as in Table 3.3.

Table 3.3 selects entropy pairs with one and two nonzero entropies in Table 3.2. The upper part and lower part based on dotted line are entropy pairs with one and two nonzero entropies, respectively. For $i = 2, 3, 4372, 4373$, there is only one nonzero entropy when the neighborhood size is 15. That means when the neighborhood size is under 13, all nearest neighbors belong to one class, and some elements appear in the other class when the neighborhood size is 15. For $i = 4, 5, .., 9, 4366, 4367, ..., 4371$, there are two nonzero entropies when the neighborhood sizes are 13 and 15. It indicates all nearest neighbors belong to one class until the neighborhood size is under 11, and some elements appear in the other class when the neighborhood sizes are 13 and 15. Equally with Figure 3.3, we can separately construct a scatterplot with entropy pairs which have same number of nonzero entropies. Figure 3.5 marks the points with the same number of nonzero entropies from one to six. From Figure 3.5, points with the same number of nonzero entropies belong to the same line. Therefore, the number of nonzero entropies will contribute to the analysis of scatterplots.

Table 3.3: Enumeration of nearest neighbors entropy with one and two nonzero values (Note that: N_i^+ and N_i^- are written in parentheses)

i	$H_{i,1}$	$H_{i,3}$	$H_{i,5}$	$H_{i,7}$	$H_{i,9}$	$H_{i,11}$	$H_{i,13}$	$H_{i,15}$	μ_i	σ_i
2	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0 (0, 13)	0.2449 (1, 14)	0.0306	0.0866
3	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0 (0, 13)	0.3927 (2, 13)	0.0491	0.1388
4372	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0 (11, 0)	0 (13, 0)	0.3927 (13, 2)	0.0491	0.1388
4373	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0 (11, 0)	0 (13, 0)	0.2449 (14, 1)	0.0306	0.0866
4	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.2712 (1, 12)	0.2449 (1, 14)	0.0645	0.1197
5	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.2712 (1, 12)	0.3927 (2, 13)	0.0830	0.1570
6	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.2712 (1, 12)	0.5004 (3, 12)	0.0964	0.1888
7	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.4293 (2, 11)	0.3927 (2, 13)	0.1027	0.1905
8	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.4293 (2, 11)	0.5004 (3, 12)	0.1162	0.2160
9	0 (0, 1)	0 (0, 3)	0 (0, 5)	0 (0, 7)	0 (0, 9)	0 (0, 11)	0.4293 (2, 11)	0.5799 (4, 11)	0.1262	0.2370
4366	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0 (11, 0)	0.4293 (11, 2)	0.5799 (11, 4)	0.1262	0.2370
4367	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0 (11, 0)	0.4293 (11, 2)	0.5004 (12, 3)	0.1162	0.2160
4368	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0 (11, 0)	0.4293 (11, 2)	0.3927 (13, 2)	0.1027	0.1905
4369	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0 (11, 0)	0.2712 (12, 1)	0.5004 (12, 3)	0.0964	0.1888
4370	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0 (11, 0)	0.2712 (12, 1)	0.3927 (13, 2)	0.0830	0.1570
4371	0 (1, 0)	0 (3, 0)	0 (5, 0)	0 (7, 0)	0 (9, 0)	0 (11, 0)	0.2712 (12, 1)	0.2449 (14, 1)	0.0645	0.1197

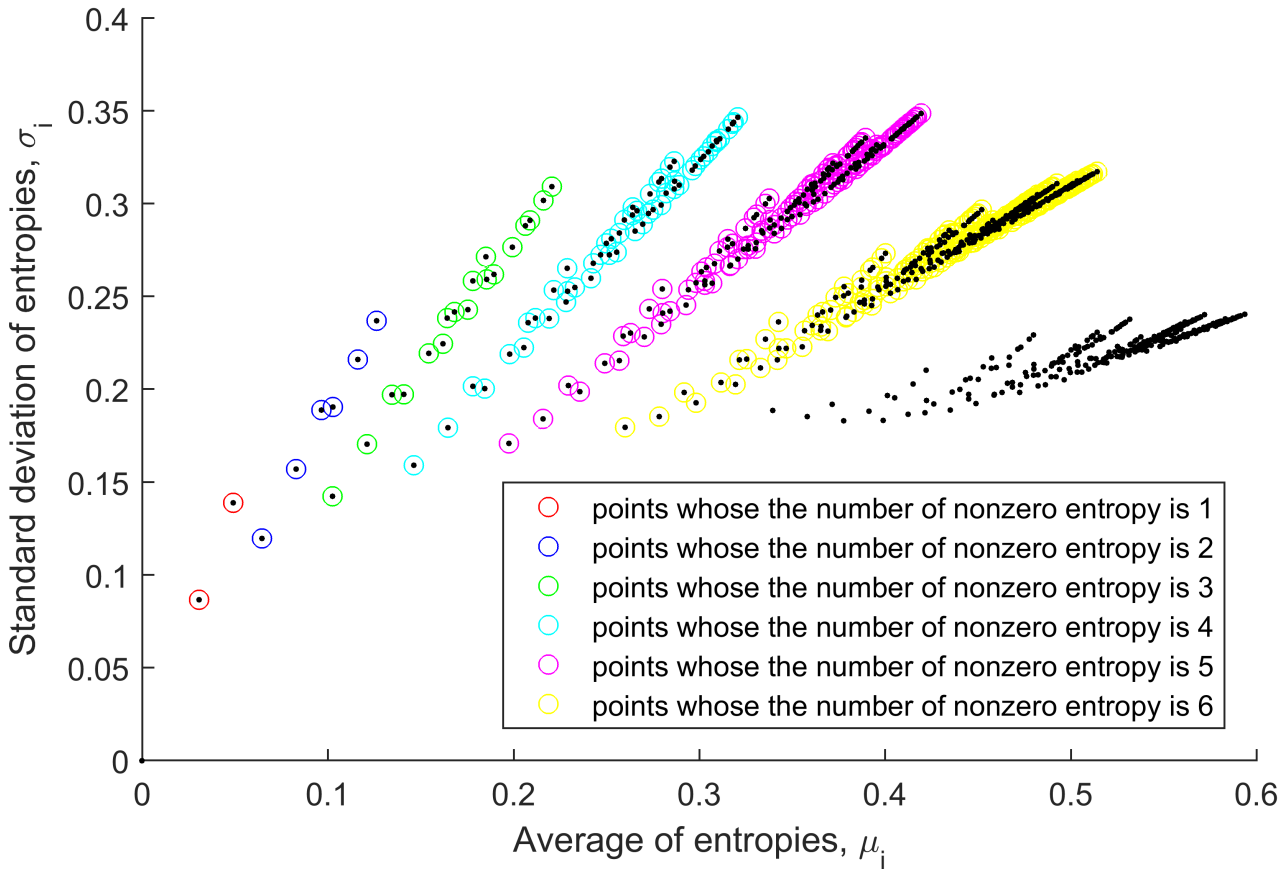

Figure 3.5: Scatterplot depending on the number of nonzero entropy

3.3 Interpretation of graphical pattern

In Figure 3.5, the set of lines can be characterized by the number of nonzero entropy, and we infer that these lines may fit well with the polar coordinates. Then, we transform the scatterplot (μ_i, σ_i) into (d_i, θ_i) as in Figure 3.6. As expected, many points are distributed in the same θ_i. To analyze this graphical pattern, we construct the following two situations.

The first is fixed θ_i and increase of d_i. In this case, the number of nonzero entropy is fixed. When d_i increases, μ_i and σ_i proportionally increase as in Figure 3.5. Increase of μ_i indicates increase of entropy, and the information is uncertain. Increase in σ_i shows that the components of entropy pairs highly vary by the neighborhood size, then the information is also uncertain.

The second is increment of θ_i. When θ_i increases, d_i proportionally increases as in Figure 3.6. As mentioned previously, increment of d_i means that the information is uncertain. Also, increase of θ_i incurs an increase in the number of nonzero entropies. As the number of nonzero entropies increases, both μ_i and σ_i increase as a whole. Then, it also shows that the information is uncertain.

Thus, increments of d_i and θ_i tend to cause uncertainty of information. Through the graphical pattern of nearest neighbors entropy, we have discovered more practical usages of the entropy, and then it can lend aid to the development of a novel fuzzy membership evaluation.

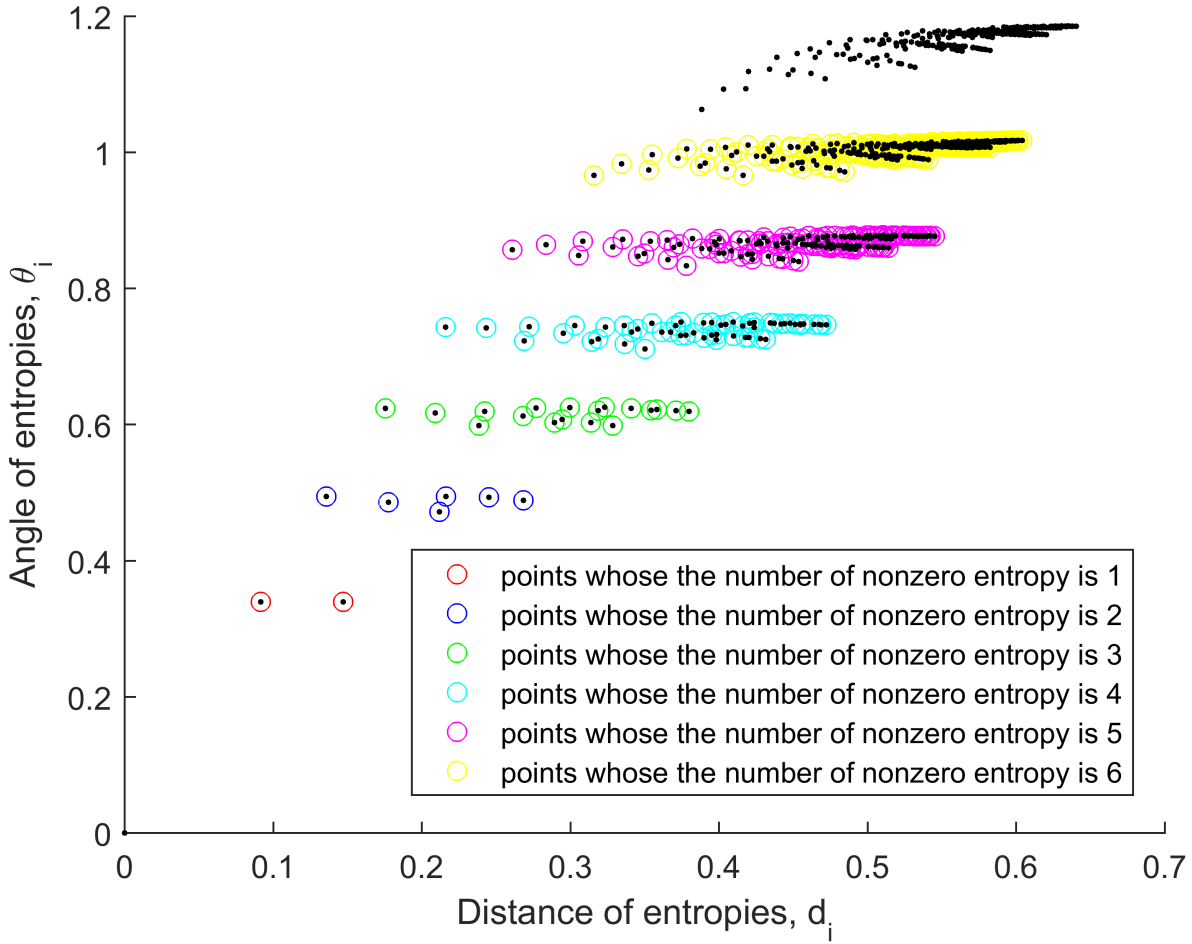

Figure 3.6: Polar coordination of the scatterplot

Chapter 4

Instance-based entropy fuzzy support vector machine

4.1 Instance-based entropy fuzzy support vector machine

4.1.1 Fuzzy support vector machine

Support vector machine (SVM) is a classifier deciding the optimal separating hyperplane with the largest margin, and solves an optimization problem as follows [65]. Let the training set be $S = \{(x_i, y_i) : i = 1, ..., N\}$, x_i be n-dimensional sample, and $y_i \in \{-1, 1\}$ for binary classification problem.

$$\begin{aligned} &\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i \\ &s.t. \quad y_i(\langle w, \phi(x_i) \rangle + b) \geq 1 - \xi_i, \\ &\qquad \xi_i \geq 0, \quad with \quad i = 1, 2, ..., N \end{aligned} \tag{4.1}$$

where w is the weight vector of decision surface, C is the regularization parameter tuned by the parameter selection, ξ_i is the slack variables to relax the margin, $\phi(x)$ is the non-linear feature mapping, and b indicates the bias.

Note that C is the only free parameter among the parameters of SVM, then it should be tuned to balance between the classification violation and margin maximization [7].

FSVM can assign the importance of each data differently to determine the decision surface. This feature is well suited to imbalanced classification. In imbalanced classification, the data in minority class is generally more important than the data in majority class and should be better classified. Therefore, if the importance of minority class is set to high priority while majority class has low importance, imbalanced data can be classified effectively. The quadratic optimization equation of FSVM is as follows [7].

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} s_i\xi_i \\ &s.t. \quad y_i(\langle w, \phi(x_i)\rangle + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0, \quad 0 \leq s_i \leq 1, \quad with \quad i = 1, 2, ..., N \end{aligned} \tag{4.2}$$

where s_i is the fuzzy membership that can be used to allocate a different weight for each i. FSVM differs from SVM in that it multiplies slack variables (ξ_i) with fuzzy membership (s_i) to differently relax the margin for each i. If s_i has a value of one for all i, Eq.(4.2) is equal to Eq.(4.1) [65]. To solve the above

equation, we can transform it into below dual problem [7].

$$\begin{aligned} &\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle \\ &s.t. \quad \sum_{i=1}^{N} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq s_i C, \\ &\quad with \quad i = 1, 2, ..., N \end{aligned} \tag{4.3}$$

We employ Sequential Minimal Optimization (SMO) [95] to solve the above dual problem, and achieve the optimal values for α_i. Then, we can calculate the weight vector and the decision function as follows [96].

$$w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i), \quad f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i \langle \phi(x_i), \phi(x_j) \rangle + b) \tag{4.4}$$

4.1.2 Entropy fuzzy membership

Entropy can be a measure of information's certainty, and therefore there have been some studies to formulate fuzzy membership through nearest neighbors entropy [10, 62]. For example, a high entropy indicates unclear information, which means that the entropy of instance is not helpful to classification, so the fuzzy membership can be assigned low. On the other hand, a low entropy indicates certain information, which means that the entropy of instance is useful to classification, so the fuzzy membership can be arranged high. Therefore, there is a negative relation between nearest neighbors entropy and fuzzy member-

ship. The following equation can be an example of entropy fuzzy membership, s_i, considering the negative relation [10].

$$s_i = \begin{cases} 1 & \text{if } y_i = +1 \\ (1 - H_i)/IR & \text{if } y_i = -1. \end{cases} \tag{4.5}$$

where H_i indicates nearest neighbors entropy in binary classification as in Eq.(3.1), and $IR = n_{maj}/n_{min}$. Note that n_{maj} and n_{min} denote the number of samples in the majority and minority class, respectively. Imbalanced ratio, IR, is the scale of how imbalanced the number of samples is. Then, we divide IR to reduce the importance of majority samples. $(1 - H_i)$ term reflects the negative relation between fuzzy membership and nearest neighbors entropy. The detailed process of entropy fuzzy membership evaluation is summarized in Algorithm 1.

According to the algorithm, the input consists of the training data, kernel function, and neighborhood size, whereas the output is the fuzzy membership of each instance. Originally, FSVM returns support vectors rather than fuzzy membership, however, we omit the following process to better compare the existing EFSVM and proposed IEFSVM by writing only fuzzy membership evaluation process. In detail of the algorithm, we first have to tune the number of nearest neighbors. For validation data, we perform 5-fold cross-

Algorithm 1 Entropy fuzzy membership evaluation

Input : Training data $X = \{(x_i, y_i)\}_{i=1}^{N}$, $y_i \in \{+1, -1\}$, kernel function, neighborhood size k
Output : Fuzzy membership of each instance $\{s_i\}_{i=1}^{N}$
procedure TUNE THE NUMBER OF NEAREST NEIGHBORS
    for $k = 1, 3, 5, 7, 9, 11, 13, 15$ **do**
        $Pt \leftarrow$ 5-fold of training data $\{(x_i, y_i)\}_{i=1}^{N}$
        for $n = 1$ to 5 **do**
            $Val \leftarrow n$th samples from Pt
            $Train \leftarrow$ remainder from Pt
            for $i = 1$ to length($Train$) **do**
                if $y_i = -1$ **then**
                    Search k nearest neighbors for each sample i in $Train$
                    Evaluate H_i for each sample i in $Train$ by Eq.(3.1)
                end if
                Evaluate s_i for each sample i in $Train$ by Eq.(4.5)
            end for
            $Mdl \leftarrow$ Fit FSVM with s_i
            Predict Val with Mdl
        end for
        $Error_k \leftarrow$ 5-fold cross-validation error of FSVM with s_i
    end for
    $k_{opt} \leftarrow \text{argmin}_k Error_k$
end procedure
procedure EVALUATE FUZZY MEMBERSHIP WITH TUNED NEIGHBORHOOD SIZE
    for $i = 1$ to N **do**
        if $y_i = -1$ **then**
            Search k_{opt} nearest neighbors for each sample i in X
            Evaluate H_i for each sample i in X by Eq.(3.1)
        end if
        Evaluate s_i for each sample i in X by Eq.(4.5)
    end for
Return $\{s_i\}_{i=1}^{N}$
end procedure

validation to achieve the optimal neighborhood size. We determine k_{opt} whose cross-validation error is the lowest. Then, we can evaluate the fuzzy membership with optimized neighborhood size, k_{opt}. Therefore, this entropy fuzzy membership varies with the neighborhood size, k. A small k denotes that the entropies are calculated close to the sample, which can result in overfitting. On the contrary, a large k indicates that the entropies are obtained away from the sample, which can ignore the small volumes of information and make a complex distribution difficult to deal with. Therefore, it is important to set the appropriate neighborhood size that matches each dataset.

4.1.3 Instance-based entropy fuzzy membership

In order to improve EFSVM which employs unified neighborhood size for all data, we consider the combination of information from all neighborhood size. Based on the graphical pattern of nearest neighbors entropy in section 3.2 and 3.3, we can consider the following four logics for decision of s_i. First, s_i should be decreased if μ_i increases, since fuzzy membership and entropy show a negative relation. Second, s_i should be reduced if σ_i increases, since increase in σ_i indicates that the components of entropy pairs highly vary by the neighborhood size, and then information is uncertain. Thirdly, if θ_i increases, s_i should be decreased, because increase of θ_i demonstrates increase in both μ_i and σ_i by section 3.3. Lastly, if d_i increases, s_i should be reduced, since an

increment of d_i also indicates increase in both μ_i and σ_i by section 3.3.

On the basis of suggested four logics, we propose an instance-based entropy fuzzy membership as follows [30].

$$s_i = \begin{cases} 1 & \text{if } y_i = +1 \\ (1 - \frac{d_i\theta_i - \min_i d_i\theta_i}{\max_i d_i\theta_i - \min_i d_i\theta_i})/IR & \text{if } y_i = -1. \end{cases} \tag{4.6}$$

where $\min_i d_i\theta_i$ and $\max_i d_i\theta_i$ indicate the minimum and maximum values of $d_i\theta_i$, respectively. When constructing a training set, not all the samples are used at once. The experiment randomly selects 1000 samples at a time, and 600 samples corresponding to 60% become the training sets. At this time, if we plot the scatterplot as in Figure 3.4 only with the 600 samples, it will be a subset of the points in Figure 3.4, then the points for d_i and θ_i with the training set cannot represent all the points in Figure 3.4. Thus, the maximum and minimum values of $d_i \times \theta_i$ for those 600 samples will vary according to each training set. Also, we visualize this fuzzy membership considering the level curve in Figure 4.1. This curve is $d_i \times \theta_i = c$ for $c = 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75$.

The detailed process of instance-based entropy fuzzy membership evaluation is summarized in Algorithm 2. Equal to EFSVM, the input consists of the training data, kernel function, and neighborhood size, whereas the output is the fuzzy membership of each instance. Originally, FSVM returns support

vectors rather than fuzzy membership, however, we omit the following process to better compare the existing EFSVM and proposed IEFSVM by writing only fuzzy membership evaluation process. According to the algorithm, there is no need to tune the number of nearest neighbors. We just evaluate d_i and θ_i for each sample i to calculate the fuzzy membership. Therefore, the proposed IEFSVM can reflect information from all neighborhood sizes efficiently, and reduce the learning time by eliminating the tuning process.

Actually, the run-time of SVMs is considered slower than other algorithms with similar classification performance [97]. In particular, when the number of sample is very large, the situation gets worse. However, the learning time of IEFSVM is fast among the SVM-based classifiers. Both the existing EFSVM and proposed IEFSVM are composed of two steps, which collects k nearest neighbors of each sample and trains the classifiers. Since the procedure of collecting k nearest neighbors of each sample is same process for both classifiers, the complexity of procedure is identical, $O(kNN_{neg})$ [10], where k is the neighborhood size, N denotes the number of training samples, and N_{neg} indicates the number of negative samples. For the minority samples, the fuzzy membership is assigned to be 1, then the collecting procedure is not needed. IEFSVM calculate the support vector only once, whereas EFSVM should train the model as many neighborhood sizes as the model sets to tune the number of nearest neighbors. Therefore, the time to train IEFSVM is shorter than that

of EFSVM by the neighborhood sizes. Since IEFSVM involves searching the nearest neighbors, and the complexity of SVM is $O(N^2)$, IEFSVM will take about twice as long as SVM when learning the model. However, there are many differences in classification performance between the two.

Algorithm 2 Instance-based entropy fuzzy membership evaluation

Input : Training data $X = \{(x_i, y_i)\}_{i=1}^{N}$, $y_i \in \{+1, -1\}$, kernel function, neighborhood size k
Output : Fuzzy membership of each instance $\{s_i\}_{i=1}^{N}$
procedure EVALUATE FUZZY MEMBERSHIP
    for $i = 1$ to N **do**
        if $y_i = -1$ **then**
            for $k = 1, 3, 5, 7, 9, 11, 13, 15$ **do**
                Search k nearest neighbors for each sample i in X
                Evaluate H_i for each sample i in X by Eq.(3.1)
            end for
            Evaluate μ_i and σ_i for each sample i in X by Eq.(3.3)
            Evaluate d_i and θ_i for each sample i in X by Eq.(3.4)
        end if
        Evaluate s_i for each sample i in X by Eq.(4.6)
    end for
Return $\{s_i\}_{i=1}^{N}$
end procedure

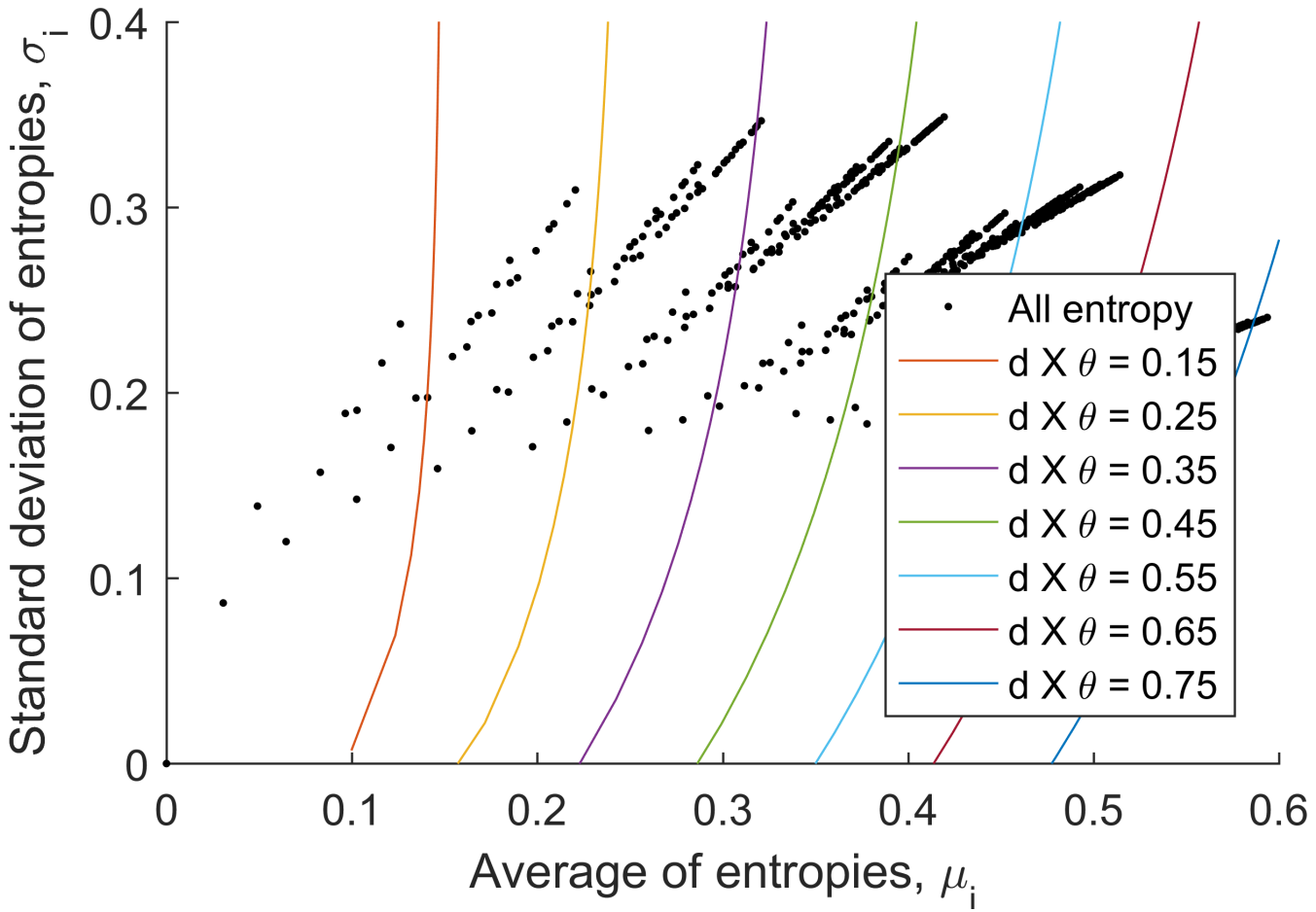

Figure 4.1: Polar coordination of the scatterplot

4.2 Experiments with public imbalanced datasets

To demonstrate the effectiveness of proposed IEFSVM, we first employ public datasets to compare with other classifiers. These public datasets are obtained from UCI [98] and we use a total of 35 datasets with imbalanced ratio from 1.14 to 15.46. Other learning machines are divided into two classes: SVM as a basic classifier and the other. This is because the proposed method is based on SVM, and many datasets have different classification performance depending on what the underlying classifier is. This is because the distribution of each data is different and there are underlying classifiers suitable for the distribution. Nevertheless, IEFSVM can be said to be appropriate for imbalanced classification if the proposed IEFSVM outperforms the algorithms that use other underlying classifier. Also, it is necessary to compare with the algorithms that use SVM as a basic classifier. Therefore, we divide the learning machine into two types for more reliable comparison. SVM-based classifiers consist of canonical SVM [65], SVM with undersampling (u-SVM), cost-sensitive SVM (cs-SVM), FSVM [52], and EFSVM [10], whereas six other algorithms are comprised of cost-sensitive AdaBoost (cs-AdaBoost) [66], cost-sensitive Random Forest (cs-RF) [67], EasyEnsemble [68], RUSBoost [69], weighted-ELM [70], and cost-sensitive XGBoost (cs-XGBoost) [34].

4.2.1 Datasets and setup

Among the datasets with imbalanced characteristics, the most representative 35 datasets are selected from the UCI repository [98]. Table 4.1 shows the description of the datasets, and the datasets are sorted in ascending order according to IR values. The first column indicates the dataset, and it consists of data name and class number. If there is only number without 'vs', the number means the minority class, and the remainders are the majority class. For example, in the case of 'dermatology456', the name of dataset in UCI is 'dermatology', and the total data of class 4, 5, and 6 are minority compared to the remainder. Meanwhile, if there is 'vs' between two numbers, the first and the second number represent the minority and the majority classes, respectively. In this case, we do not use the whole data, but only employ listed two classes. For example, in the case of 'liver2vs1', the name of dataset in UCI is 'liver', and the second and the first classes of 'liver' are the minority and the majority classes, respectively. The second column represents the sorted IR, the third column indicates the number of whole instances, the fourth and fifth column show the number of the minority and the majority samples, respectively, and the sixth column is the dimension of each dataset.

Comparing learning machines are composed of five SVM-based learning algorithms and six other algorithms. We distinguish the two parts and present

Table 4.1: Description of imbalanced UCI datasets

Dataset	IR	Inst.	Pos.	Neg.	Dim.
ecoli8	5.46	336	52	284	7
dermatology4	6.46	358	48	310	34
dermatology5	6.46	358	48	310	34
ecoli5	8.6	336	35	301	7
ecoli178vs46	9.09	222	22	200	7
ecoli1345vs6	9.1	202	20	182	7
ecoli157vs6	9.15	203	20	183	7
ecoli12vs346	9.17	244	24	220	7
ecoli1378vs46	9.18	224	22	202	7
glass15vs6	9.22	92	9	83	9
ecoli1457vs6	9.25	205	20	185	7
zoo1257vs6	9.38	83	8	75	16
ecoli178vs6	10	220	20	200	7
abalone9vs16	10.28	756	67	689	8
ecoli12vs6	11	240	20	220	7
abalone10vs4	11.12	691	57	634	8
zoo6	11.63	101	8	93	16
abalone9vs17	11.88	747	58	689	8
abalone9vs4	12.09	746	57	689	8
ecoli1257vs6	13	280	20	260	7
glass5	15.46	214	13	201	9

each result of the comparison. SVM-based learning algorithms such as SVM, u-SVM, cs-SVM, FSVM, EFSVM, and IEFSVM employ the radial basis function (RBF) kernel or linear kernel. The regularization parameter C is selected from $\{2^{-6}, 2^{-4}, ..., 2^4, 2^6\}$. Entropy-based learning machines such as EFSVM and IEFSVM choose neighborhood size from $\{1, 3, 5, 7, 9, 11, 13, 15\}$ when calculating the nearest neighbors entropy. Tree-based learning algorithms such as cs-AdaBoost, cs-RF, EasyEnsemble, and RUSBoost employ 100 as the maximum value of learning iterations, while cs-XGBoost conforms to the tuning technique in Xia *et al.* [34] and Jain [99]. All tuning procedures follow a 5-fold cross validation.

To evaluate the performance of imbalanced classification, we employ AUC, which is a significant tool for measuring the imbalanced classification performance [11, 76], and AUC is defined as follows.

$$AUC = (1 + TP_{rate} - FP_{rate})/2. \tag{4.7}$$

where TP_{rate} and FP_{rate} indicate the proportion of positive samples correctly classified and that of negative samples misclassified, respectively.

4.2.2 Results

Our proposed fuzzy membership evaluation is compared with two types of benchmark, five SVM-based algorithms and six other algorithms. First, Table 4.2 shows the mean and standard deviation of AUC values of SVM-based learning machines on UCI datasets with 100 experiments. It is sorted in ascending order according to IR, and the best results are highlighted in bold. In fact, the proposed IEFSVM shows good performance when IR has a high value, then we observe the performance of each learning machine for the datasets with IR over 5. As a result, IEFSVM obtains the seven highest AUC among 21 datasets, and also have the highest average AUC of 93. Specifically, Table 4.3 demonstrates the rankings of AUC values in Table 4.2. In the same manner, IEFSVM outperforms other SVM-based algorithms for highly imbalanced datasets. In order to demonstrate the effectiveness of instance-based procedure, better results between existing EFSVM and proposed IEFSVM for all datasets are highlighted in bold. In fact, IEFSVM is better than EFSVM for 18 of 21 highly imbalanced datasets.

Table 4.2: AUC values with SVM based learning machines on UCI datasets

Dataset	IR	SVM	u-SVM	cs-SVM	FSVM	EFSVM	IEFSVM
ecoli8	5.46	91.73±3.47	91.01±3.7	92.82±2.35	92.82±2.65	92.86±2.59	**93.1±2.59**
dermatology4	6.46	93.49±3.13	92.58±2.73	94.7±2.82	95.04±2.54	95.03±2.81	**95.57±2.17**
dermatology5	6.46	98.96±2.3	99.5±0.97	99.76±0.68	**99.78±0.54**	99.49±1.19	99.73±0.67
ecoli5	8.6	79.74±6.57	87.14±3.52	**87.84±3.47**	87.77±3.82	87.62±3.85	87.77±3.4
ecoli178vs46	9.09	84.62±6.51	89.14±5.12	87.74±5.19	**89.76±5.39**	89.16±5.04	89.17±5.05
ecoli1345vs6	9.1	94.44±4.4	96.75±2.61	96.55±3.23	96.04±3.6	96.35±3.16	**97.2±2.7**
ecoli157vs6	9.15	93.02±6.12	**95.99±2.9**	94.98±3.42	94.1±3.48	94.71±3.34	94.97±2.9
ecoli12vs346	9.17	89.5±6.34	91.97±4.28	**92.91±4.83**	92.56±5.47	92.38±5.05	92.47±4.32
ecoli1378vs46	9.18	85.46±6.17	87.7±4.94	87.51±4.85	86.95±5.53	87.75±5.24	**88.05±4.7**
glass15vs6	9.22	83.67±10.97	83.32±11.4	84.52±10.5	87.39±10.5	86.14±10.79	**89.83±8.63**
ecoli1457vs6	9.25	94.42±5.19	**95.68±3.92**	94.65±3.87	94.89±3.66	94.95±3.65	95.48±3.71
zoo1257vs6	9.38	96.7±7.33	90.63±8.23	96.47±6.53	96.73±7.77	97.03±5.72	**97.27±7.03**
ecoli178vs6	10	90.11±6.02	**92.62±3.83**	91.23±4.4	90.51±5.15	92.39±4.86	92.49±4.13

Table 4.2 – continued from previous page

Dataset	IR	SVM	u-SVM	cs-SVM	FSVM	EFSVM	IEFSVM
abalone9vs16	10.28	74.63±3.5	82.42±4.67	83.7±3.8	83.93±3.59	**84.1±3.29**	83.86±4.21
ecoli12vs6	11	94.92±3.94	**97.27±2.83**	95.5±3.4	95.7±3.71	96.3±3.81	96.2±3.05
abalone10vs4	11.12	97.55±2.38	97.41±1.98	97.59±1.08	**97.68±1.01**	97.45±1.3	97.55±0.84
zoo6	11.63	94.35±12.02	93.78±5.09	95.7±8.97	95.5±9.78	96.7±7.77	**98.51±1.34**
abalone9vs17	11.88	71.39±4.55	82.28±4.91	**84.36±4.2**	83.71±4.11	84.01±3.53	84.17±3.84
abalone9vs4	12.09	96.7±2.3	**97.67±1.16**	97.52±1.37	97.59±1.21	97.31±1.36	97.45±1.01
ecoli1257vs6	13	93.54±5.3	**96.22±3.42**	93.26±4.98	92.93±5.41	94.19±3.68	95±4.29
glass5	15.46	77.26±9.76	83.86±9.83	85.24±8.75	87.46±7.38	**89.33±5.79**	87.08±9.17
Average AUC		89.34±5.63	91.66±4.38	92.12±4.41	92.33±4.59	92.63±4.18	**93±3.8**

Table 4.3: AUC rankings with SVM based learning machines on UCI datasets

Dataset	IR	SVM	u-SVM	cs-SVM	FSVM	EFSVM	IEFSVM
ecoli8	5.46	5	6	4	3	2	**1**
dermatology4	6.46	5	6	4	2	3	**1**
dermatology5	6.46	6	4	2	1	5	**3**
ecoli5	8.6	6	5	1	3	4	**2**
ecoli178vs46	9.09	6	4	5	1	3	**2**
ecoli1345vs6	9.1	6	2	3	5	4	**1**
ecoli157vs6	9.15	6	1	2	5	4	**3**
ecoli12vs346	9.17	6	5	1	2	4	**3**
ecoli1378vs46	9.18	6	3	4	5	2	**1**
glass15vs6	9.22	5	6	4	2	3	**1**
ecoli1457vs6	9.25	6	1	5	4	3	**2**
zoo1257vs6	9.38	4	6	5	3	2	**1**
ecoli178vs6	10	6	1	4	5	3	**2**

Table 4.3 – continued from previous page

Dataset	IR	SVM	u-SVM	cs-SVM	FSVM	EFSVM	IEFSVM
abalone9vs16	10.28	6	5	4	2	**1**	3
ecoli12vs6	11	6	1	5	4	**2**	3
abalone10vs4	11.12	3	6	2	1	5	**4**
zoo6	11.63	5	6	3	4	2	**1**
abalone9vs17	11.88	6	5	1	4	3	**2**
abalone9vs4	12.09	6	1	3	2	5	**4**
ecoli1257vs6	13	4	1	5	6	3	**2**
glass5	15.46	6	5	4	2	**1**	3
Average rank		5.48	3.81	3.38	3.14	3.05	**2.14**

Secondly, Table 4.4 indicates the mean and standard deviation of AUC values of six state-of-the-art algorithms on UCI datasets with 100 experiments. Equally, it is sorted in ascending order according to IR, and the best results are highlighted in bold. The proposed IEFSVM shows sound performance when IR has a high value, then we also demonstrate the classification performance with highly imbalance datasets. While w-ELM obtains the nine highest AUC, IEFSVM have the six highest AUC among 21 datasets. IEFSVM, however, indicates the highest average of AUC, and is much better than w-ELM. To be specific, Table 4.5 demonstrates the rankings of AUC values in Table 4.4. As a result, IEFSVM is superior to all state-of-the-art algorithms.

Table 4.4: AUC values with six state-of-the-art algorithms on UCI datasets

Dataset	IR	cs-AdaBoost	cs-RF	EasyEnsemble	RUSBoost	w-ELM	cs-XGBoost	IEFSVM
ecoli8	5.46	88.35±3.35	88.66±4.31	89.3±2.89	86.84±3.47	89.42±2.84	87.24±4.64	**93.1±2.59**
dermatology4	6.46	93.36±3.16	93.83±3.32	94.23±2.68	88.39±3.55	94.3±1.23	91.45±3.92	**95.57±2.17**
dermatology5	6.46	90.85±9.33	99.82±0.64	74.08±4.39	90.82±9.32	**99.85±0.28**	99.78±0.29	99.73±0.67
ecoli5	8.6	84.11±7.21	79.4±6.98	87.11±3.94	87.03±4.27	87.06±2.32	86.62±5.9	**87.77±3.4**
ecoli178vs46	9.09	86.24±6.37	85.83±5.83	88.24±4.51	87.92±5.21	**91.68±4.23**	88.56±6.86	89.17±5.05
ecoli1345vs6	9.1	94.77±6.09	92.85±5.86	96.81±2.76	95.29±4.66	**98.11±1.87**	93.2±5.53	97.2±2.7
ecoli157vs6	9.15	93.6±6.46	92.46±7.65	95.03±3.43	93.48±6.14	**95.95±2.52**	92.51±6.53	94.97±2.9
ecoli12vs346	9.17	87.99±5.58	86.95±7.18	88.9±4.33	87.82±5.26	**95.62±3.03**	85.87±6.06	92.47±4.32
ecoli1378vs46	9.18	85.28±5.84	86.15±5.64	86.53±5.72	86.95±4.89	**89.69±5.67**	86.62±5.15	88.05±4.7
glass15vs6	9.22	77.51±4.72	98.42±3.84	92.14±6.67	94.96±8.16	92.49±5.82	**99.03±1.05**	89.83±8.63
ecoli1457vs6	9.25	94.4±5.74	91.61±7.76	95.75±4.1	92.23±5.6	**96.58±2.36**	92.31±5.53	95.48±3.71
zoo1257vs6	9.38	92.9±1.58	97.63±5.31	87.47±8.57	95.23±5.23	**97.83±4.69**	94.72±7.07	97.27±7.03
ecoli178vs6	10	91.72±5.21	91.11±6.35	**94.04±4.33**	93.19±4.27	92.25±3.15	93.79±4.5	92.49±4.13

Table 4.4 – continued from previous page

Dataset	IR	cs-AdaBoost	cs-RF	EasyEnsemble	RUSBoost	w-ELM	cs-XGBoost	IEFSVM
abalone9vs16	10.28	77.25±3.82	72.09±4.08	79.32±3.22	72.94±3.46	68.2±5.09	78.67±3.87	**83.86±4.21**
ecoli12vs6	11	91.18±6.55	90.43±7.91	97.63±1.69	93.28±6.86	**98.7±0.75**	91.31±6.93	96.2±3.05
abalone10vs4	11.12	84.66±1.89	98.14±1.53	98.12±1.23	95.35±1.64	97.52±1.05	**98.27±1.51**	97.55±0.84
zoo6	11.63	93.3±1.71	98.2±4.76	87.05±0.37	95.28±7.98	96.51±7.82	95.31±7.33	**98.51±1.34**
abalone9vs17	11.88	78.35±4.81	72.1±4.3	83.04±3.69	68.83±4.23	67.4±6.11	79.4±4.69	**84.17±3.84**
abalone9vs4	12.09	91.38±5.58	**97.9±1.96**	97.49±1.42	97.46±2.12	97.85±2.33	97.79±2.22	97.45±1.01
ecoli1257vs6	13	91.56±7.82	89.85±8.95	**96.19±2.84**	93.55±6.67	95.9±2.4	91.04±7.45	95±4.29
glass5	15.46	80.18±1.75	83.73±0.62	**88.15±5.28**	87.73±7.35	86.16±6.2	82.29±11.43	87.08±9.17
Average AUC		88.04±9.27	89.86±5.47	90.32±5.15	89.74±6.21	91.84±3.36	90.75±5.17	**93±3.8**

Table 4.5: AUC rankings with six state-of-the-art algorithms on UCI datasets

Dataset	IR	cs-AdaBoost	cs-RF	EasyEnsemble	RUSBoost	w-ELM	cs-XGBoost	IEFSVM
ecoli8	5.46	5	4	3	7	2	6	1
dermatology4	6.46	5	4	3	7	2	6	1
dermatology5	6.46	5	2	7	6	1	3	4
ecoli5	8.6	6	7	2	4	3	5	1
ecoli178vs46	9.09	6	7	4	5	1	3	2
ecoli1345vs6	9.1	5	7	3	4	1	6	2
ecoli157vs6	9.15	4	7	2	5	1	6	3
ecoli12vs346	9.17	4	6	3	5	1	7	2
ecoli1378vs46	9.18	7	6	5	3	1	4	2
glass15vs6	9.22	7	2	5	3	4	1	6
ecoli1457vs6	9.25	4	7	2	6	1	5	3
zoo1257vs6	9.38	6	2	7	4	1	5	3
ecoli178vs6	10	6	7	1	3	5	2	4

Table 4.5 – continued from previous page

Dataset	IR	cs-AdaBoost	cs-RF	EasyEnsemble	RUSBoost	w-ELM	cs-XGBoost	IEFSVM
abalone9vs16	10.28	4	6	2	5	7	3	1
ecoli12vs6	11	6	7	2	4	1	5	3
abalone10vs4	11.12	7	2	3	6	5	1	4
zoo6	11.63	6	2	7	5	3	4	1
abalone9vs17	11.88	4	5	2	6	7	3	1
abalone9vs4	12.09	7	1	3	4	6	2	5
ecoli1257vs6	13	5	7	1	4	2	6	3
glass5	15.46	7	5	1	2	4	6	3
Average rank		5.52	4.9	3.24	4.67	2.81	4.24	2.62

4.2.3 Statistical studies

For the above AUC results, it is necessary to statistically prove that IEFSVM is better than other algorithms. We therefore perform Wilcoxon paired signed-rank test [100] and Holm post hoc test [101]. Wilcoxon paired signed-rank test is used as an alternative to paired t-test to statistically check whether the mean of two populations are different. Holm post hoc test is used for multiple comparisons, and z value of this test is as follows [102].

$$z = (R^* - R)/\sqrt{k(k+1)/(6N)} \tag{4.8}$$

where R^* denotes the average AUC ranking of IEFSVM on datasets, R indicates the average AUC ranking of other algorithms on datasets, k refers to the number of comparing algorithms, which are 6 and 7 for the two types of benchmarks, and N is the number of datasets, which is 21 in this experiment. We can evaluate z values for each algorithm, and sort the z values in descending order. The adjusted alpha of corresponding i th algorithm for Holm post hoc test is equal to $(0.05/i)$. If the adjusted alpha is greater than p value of the z value, the hypothesis that both algorithms achieve the same AUC ranking is rejected.

Table 4.6 demonstrates the results of Holm tests with SVM-based algorithms for all highly imbalanced datasets. Then, IEFSVM beats other algo-

rithms except FSVM and EFSVM. To be specific, Table 4.7 indicates the results of Wilcoxon tests with SVM-based algorithms for all highly imbalanced datasets. As a result, IEFSVM statistically outperforms all algorithms.

Table 4.6: Holm tests with SVM based learning machines on UCI datasets

Algorithm	Z	p-Value	Holm	Hypothesis
SVM	5.4762	0	0.01	Rejected
u-SVM	3.8095	0.0019	0.0125	Rejected
cs-SVM	3.3810	0.0160	0.0167	Rejected
FSVM	3.1429	0.0416	0.025	Not rejected
EFSVM	3.0476	0.0585	0.05	Not rejected

Table 4.7: Wilcoxon tests with SVM based learning machines on UCI datasets

Algorithm	Z	p-Value	Hypothesis(α=0.05)
SVM	-4.0343	0	Rejected
EFSVM	-2.6367	0.0084	Rejected
cs-SVM	-2.6187	0.0088	Rejected
u-SVM	-2.5446	0.0109	Rejected
FSVM	-1.9861	0.0470	Rejected

Table 4.8 demonstrates the results of Holm tests with six state-of-the-art algorithms for all highly imbalanced datasets. Then, IEFSVM achieves better performance than cs-XGBoost, cs-AdaBoost, cs-RF, and RUSBoost except EasyEnsemble and w-ELM. Specifically, Table 4.9 indicates the results of Wilcoxon tests with six state-of-the-art algorithms for all highly imbalanced datasets. The results of Wilcoxon tests, however, shows the same result as Holm tests for all cases. It indicates that IEFSVM is not statistically better than EasyEnsemble and w-ELM.

Table 4.8: Holm tests with six state-of-the-art algorithms on UCI datasets

Algorithm	Z	p-Value	Holm	Hypothesis
cs-AdaBoost	5.5238	0	0.0083	Rejected
cs-RF	4.9048	0	0.01	Rejected
RUSBoost	4.6667	0.0011	0.0125	Rejected
cs-XGBoost	4.2381	0.0076	0.0167	Rejected
EasyEnsemble	3.2381	0.1766	0.025	Not rejected
w-ELM	2.8095	0.3875	0.05	Not rejected

Table 4.9: Wilcoxon tests with six state-of-the-art algorithms on UCI datasets

Algorithm	Z	p-Value	Hypothesis(α=0.05)
cs-AdaBoost	-4.0356	0	Rejected
RUSBoost	-3.1317	0.0017	Rejected
cs-RF	-2.8090	0.0050	Rejected
cs-XGBoost	-2.2245	0.0261	Rejected
EasyEnsemble	-1.0415	0.2976	Not rejected
w-ELM	0.3018	0.7628	Not rejected

4.3 Experiments with real-world imbalanced datasets

To demonstrate the effectiveness of the proposed IEFSVM, in the previous section, we employ public datasets to compare with other classifiers. In this section, we use real-world datasets to make a comparison with other classifiers. As in the previous section, we divide into two types of learning machine: SVM-based and other classifiers. Also, we perform the same benchmark algorithms and set the same parameters.

4.3.1 Data sets

Among the datasets with imbalanced characteristics, 12 real-world datasets are selected. The datasets consist of AIDS [103], Cervical cancer [104], Lending Club [105], Otto group [106], and Seoul weather [107, 108]. Since the Otto group dataset has multiple classes, we transform it into seven binary imbalanced datasets. Table 4.10 shows the information of datasets, and IR has a value from 5.31 to 26.15. The first column indicates the dataset, and we specify the class number of Otto group datasets. 'weather1' is for daily raining of Seoul, and 'weather2' is for daily air pollution of Seoul. The second column represents IR, the third column indicates the number of whole instances, the fourth and fifth column show the number of the minority and majority class, respectively, and the sixth column is the dimension of each dataset. Unlike public datasets, we single out real-world datasets with many instances.

Table 4.10: Description of imbalanced real-world datasets

dataset	IR	Inst.	Pos.	Neg.	Dim.
AIDS	26.15	38,529	1,419	37,110	6
Cancer	13.84	668	45	623	31
Lending	6.63	331,879	43,480	288,399	32
Otto1	31.08	61,878	1,929	59,949	93
Otto3	6.73	61,878	8,004	53,874	93
Otto4	21.99	61,878	2,691	59,187	93
Otto5	21.59	61,878	2,739	59,139	93
Otto7	20.8	61,878	2,839	59,039	93
Otto8	6.31	61,878	8,464	53,414	93
Otto9	11.49	61,878	4,955	56,923	93
weather1	5.31	7,677	1,216	6,461	8
weather2	16.91	12,880	719	12,161	6

4.3.2 Results

As with the previous section, the comparison is performed with two types of benchmark. First, Table 4.11 shows the mean and standard deviation of AUC values of SVM-based learning machines on real-world datasets with 100 experiments. The best results are highlighted in bold. In this case, since all IRs are greater than 5, IEFSVM ranks first in 8 out of 12 datasets. AUC values of IEFSVM are also very high for the remaining 4 datasets. The existing EFSVM, on the other hand, ranks the first in 2 out of 12 datasets. To be specific, Table 4.12 demonstrates the rankings of AUC values in Table 4.11. In the same manner, IEFSVM outperforms other SVM-based algorithms. In order to demonstrate the effectiveness of instance-based procedure, better results between existing EFSVM and proposed IEFSVM for all datasets are highlighted in bold. Consequently, IEFSVM is better than EFSVM for 10 of 12 datasets.

Secondly, Table 4.13 indicates the mean and standard deviation of AUC values of six state-of-the-art algorithms on real-world datasets with 100 experiments. The best results are highlighted in bold. In this case, IEFSVM ranks first in 6 out of 12 datasets, while RUSBoost and cs-XGBoost rank first in 2 out of 12 datasets. Specifically, Table 4.14 demonstrates the rankings of AUC values in Table 4.13. In the same manner, IEFSVM outperforms six state-of-the-art algorithms.

Table 4.11: AUC values with SVM based learning machines on real-world datasets

Dataset	IR	SVM	u-SVM	cs-SVM	FSVM	EFSVM	IEFSVM
AIDS	26.15	75.09±11.13	85.89±9.47	85.84±8.92	84.72±9.78	**86.03±10.08**	85.84±9.72
Cancer	13.84	82.15±10.14	90.78±5.24	**91.91±3.55**	91.67±3.64	91.45±4.17	91.77±3.84
Lending	6.63	51.39±2.01	57.19±4.98	59.77±4.12	58.21±5.39	57.24±5.47	**60.85±2.42**
Otto1	31.08	59.28±9.51	69.53±9.35	71.08±10.6	73.16±10.9	72.69±9.63	**73.37±9.61**
Otto3	6.73	60.57±4.68	74.11±5.39	75.94±5.14	75.95±4.74	76.42±4.15	**77.15±4.21**
Otto4	21.99	53.38±4.97	66.38±8.03	67.11±8.72	67.01±9.66	69.76±8.81	**70.78±8.09**
Otto5	21.59	90.95±6.79	92.75±4.9	93.58±5.38	**95.57±3.73**	94.8±5.33	94.9±5.15
Otto7	20.80	65.9±10.57	67.81±8.19	72.52±8.4	73.6±9.01	73.49±8.8	**74.43±8.2**
Otto8	6.31	82.52±6.43	85.77±4.16	88.17±4.27	88.31±4.18	**88.78±4.41**	88.72±3.73
Otto9	11.49	79.92±7.21	82.96±6.56	85.88±6.98	87.02±6.11	87.24±5.98	**87.48±5.84**
weather1	5.31	50±0	66.42±2.09	67.05±1.82	67.82±2.26	66.95±2.88	**70.01±2.68**
weather2	16.91	62.22±7.79	76.02±3.48	75.99±4.87	75.28±3.5	75.79±5.77	**76.81±4.95**
Average		67.78±13.27	76.3±10.76	77.9±10.49	78.19±10.79	78.39±10.82	**79.34±9.86**

Table 4.12: AUC rankings with SVM based learning machines on real-world datasets

Dataset	IR	SVM	u-SVM	cs-SVM	FSVM	EFSVM	IEFSVM
AIDS	26.15	6	2	3	5	**1**	4
Cancer	13.84	6	5	1	3	4	**2**
Lending	6.63	6	5	2	3	4	**1**
Otto1	31.08	6	5	4	2	3	**1**
Otto3	6.73	6	5	4	3	2	**1**
Otto4	21.99	6	5	3	4	2	**1**
Otto5	21.59	6	5	4	1	3	**2**
Otto7	20.8	6	5	4	2	3	**1**
Otto8	6.31	6	5	4	3	**1**	2
Otto9	11.49	6	5	4	3	2	**1**
weather1	5.31	6	5	3	2	4	**1**
weather2	16.91	6	2	3	5	4	**1**
Average rank		6	4.5	3.25	3	2.75	1.5

Table 4.13: AUC values with six state-of-the-art algorithms on real-world datasets

Dataset	IR	cs-AdaBoost	cs-RF	EasyEnsemble	RUSBoost	w-ELM	cs-XGBoost	IEFSVM
AIDS	26.15	87.68±7.74	85.34±9.51	90.48±5.6	**90.7±6.55**	75.94±9.81	90.37±5.48	85.84±9.72
Cancer	13.84	91.05±4.39	85.25±7.04	90.73±3.14	91.91±3.55	85.54±4.77	**92.37±2.7**	91.77±3.84
Lending	6.63	57.86±3.84	52.37±2.34	55.31±7.35	58.6±3.05	53.72±3.01	54.98±3.81	**60.85±2.42**
Otto1	31.08	63.04±9.78	52.78±5.49	68.99±10.96	67.08±9.53	64.4±7.55	67.27±5.62	**73.37±9.61**
Otto3	6.73	71.51±5.63	64.24±5.79	76.45±4.4	68.63±6.18	74.45±6.34	72.75±3.93	**77.15±4.21**
Otto4	21.99	64.32±7.62	54.01±5.3	**74.39±7.85**	69.3±8.29	71.19±7.12	69.27±5.93	70.78±8.09
Otto5	21.59	92.26±8.16	86.04±8.67	93.77±7.46	94.18±5.38	72.33±7.91	93.58±3.03	**94.9±5.15**
Otto7	20.80	69.06±8.99	55.8±7	72.38±8.9	**74.45±8.03**	69.41±8.88	71.99±5.38	74.43±8.2
Otto8	6.31	83.23±3.88	82.36±4.19	88.59±3.32	74.89±5.17	79.09±7.54	**88.75±2.58**	88.72±3.73
Otto9	11.49	79.04±5.95	72.14±6.58	83.94±5.21	79.01±7.39	81.7±7.66	78.72±4.06	**87.48±5.84**
weather1	5.31	62±3.85	69.95±2.61	69.51±2.27	61.24±3.62	66.51±2.02	67.87±3.22	**70.01±2.68**
weather2	16.91	**78.06±4.96**	65.93±3.07	73.09±7.74	76.08±4.5	76.92±4.61	74.44±3.99	76.81±4.95
Average		74.93±11.46	68.85±12.85	78.14±11	75.51±11.21	72.6±8.16	76.86±4.14	**79.34±9.86**

Table 4.14: AUC rankings with six state-of-the-art algorithms on real-world datasets

Dataset	IR	cs-AdaBoost	cs-RF	EasyEnsemble	RUSBoost	w-ELM	cs-XGBoost	IEFSVM
AIDS	26.15	4	6	2	1	7	3	5
Cancer	13.84	4	7	5	2	6	1	3
Lending	6.63	3	7	4	2	6	5	1
Otto1	31.08	6	7	2	3	5	4	1
Otto3	6.73	5	7	2	6	3	4	1
Otto4	21.99	6	7	1	4	2	5	3
Otto5	21.59	5	6	3	2	7	4	1
Otto7	20.8	6	7	3	1	5	4	2
Otto8	6.31	4	5	3	7	6	1	2
Otto9	11.49	4	7	2	5	3	6	1
weather1	5.31	6	2	3	7	5	4	1
weather2	16.91	1	7	6	4	2	5	3
Average rank		4.5	6.25	3	3.67	4.75	3.83	2

Overall, by experimental studies, the classification performance of our model is verified with two types of benchmarks, that is, SVM-based algorithms and other algorithms. While IEFSVM is superior to other classifiers except w-ELM and EasyEnsemble for public highly imbalanced datasets, IEFSVM outperforms all classifiers for real-world highly imbalanced datasets. In that we select the real-world highly imbalanced datasets with many instances, IEFSVM has strengths and is likely to be applied in other areas. Since the existing EFSVM uses a uniform neighborhood size for all data, it can obtain a sound classification performance if there is not much change in entropy according to the data point. On the other hand, because the proposed IEFSVM considers the combination of entropy, we can acquire great classification performance if there are many entropy changes according to the data points. The weight of minority data is not different between EFSVM and IEFSVM, on the contrary, there is a difference for the majority data. If the imbalance ratio is very high, the weight of minority data is much higher than the majority data, so that the majority data is widely distributed regardless of the decision surface. As a result, the entropy change is very large depending on where the majority data is spread, and thus the proposed classifier will have a higher imbalanced classification performance. Also, it will be better classified if the majority data exceeds the decision surface and the entropy value widely varies. The proposed algorithm could better reflect information through the combination of entropy and there-

fore show better results when visualization does not seem to be well classified. That would usually be the case when the IR is high.

Chapter 5

Investment decision in P2P lending market

5.1 Data description

Lending Club acts as a mediator between the borrower and investor, and transparently discloses borrower's personal and financial information. The investor can decide which loans to invest in through this open data, and this study employs data from Lending Club for three-year loans from 2007 to 2014. The status of loans is divided into fully paid and default, which can be assumed to be a classification problem.

5.1.1 LC grade

Not all borrower applications are accepted. LC has developed an algorithm that evaluates the credit of each borrower through existing data, and does not allow loans with low credit. Also, LC assesses the credit of allowed loans. This is called the LC grade, and it consists of 7 grades from A to G. The higher the rating, the higher the borrower's credit and the lower the interest rate. The loan statistics of the LC grade are shown in Table 5.1.

Table 5.1: Loan statistics of LC grade

Grade	A	B	C	D	E	F+G	All
Loan amount($\times 10^8$\$)	9.56	14.60	9.86	4.90	1.34	0.33	40.58
Interest rate(%)	7.52	11.52	14.57	17.61	20.46	23.75	12.64
Historical return(%)	7.12	9.83	10.05	10.62	10.68	11.59	9.45
Standard deviation(%)	13.25	18.92	24.44	29.15	33.27	37.59	21.92

Table 5.1 demonstrates loan amount, interest rate, historical return, and standard deviation of the return for all loans according to LC grade. The grade that occupies the largest loan amount of 1.460 billion dollars is B, whereas the loan amount of E, F, and G grades of relatively low credit is small. The interest rate gradually increases from A to G. The borrower of grade A can receive a loan at an interest rate of 7.52%, while G-rating borrower accepts the rate of 23.75%, which is more than three times the A-rating. The historical return is the investment return that can be obtained by investing in each grade, and it is always lower than the interest rate since there are borrowers that cannot pay back the loan. In the case of grade A, since most borrowers are highly creditable, most loans are fully paid, so that there is only 0.4% difference between the interest rate and historical return. In the case of grade F+G, the difference between the two is 12.16%. This is because loans with lower ratings have more defaults. This can be also seen from the standard deviation

of return. The higher the rating, the higher the standard deviation, then the greater the risk. The standard deviation of grade F+G is three times higher than that of the grade A. In this regard, we can observe that the standard deviation is higher for all cases than the historical return, which means that Sharpe ratio is less than 1. If we can detect the loans to be fully paid at a lower grade, we can greatly improve the portfolio return.

5.1.2 Imbalanced characteristics of loan status

Loan status is divided into fully paid and default, while the number of fully paid loans is much more than that of default loans. Assuming the problem of predicting the loan status, this is an imbalanced classification problem. In detail, the histogram of investment return is shown in Figure 5.1. The above histogram is for the entire sample, while the samples of positive return are much more than those of negative return. The histogram below is for samples whose return is negative. The 331,878 loans used in this study consist of 288,398 fully paid loans of 86.9% and 43,480 default loans of 13.1%. In terms of whether return is positive or negative, there are 292,851 loans with positive return of 88.2% and 39,017 loans with negative return of 11.8%. Thus, the loan status of LC has an imbalanced characteristic. In this manner, if we set the aim of this study as a classification problem to predict the loan status, the problem is an imbalanced classification problem considering that the number of fully

paid loans is much more than that of default loans.

5.1.3 Variables

The response variable of LC data is divided into two types. The first is loan status, which is used as an imbalanced classification model that predicts whether the loan status would be fully paid or default. The second is investment return, which is employed as a regression model to predict the investment return. LC provides many independent variables for loans and borrowers such as annual income, employment length, interest rate, number of open account, and revolving utilization rate. Then, we choose the variables that can be used in the model as follows. First, the numerical variable is normalized such that the maximum value is 1 and the minimum value is -1. Secondly, categorical variables are converted into dummy variables. Then, we use gradient boosting method [109] to measure the importance of variables. Finally, we can choose variables with high importance.

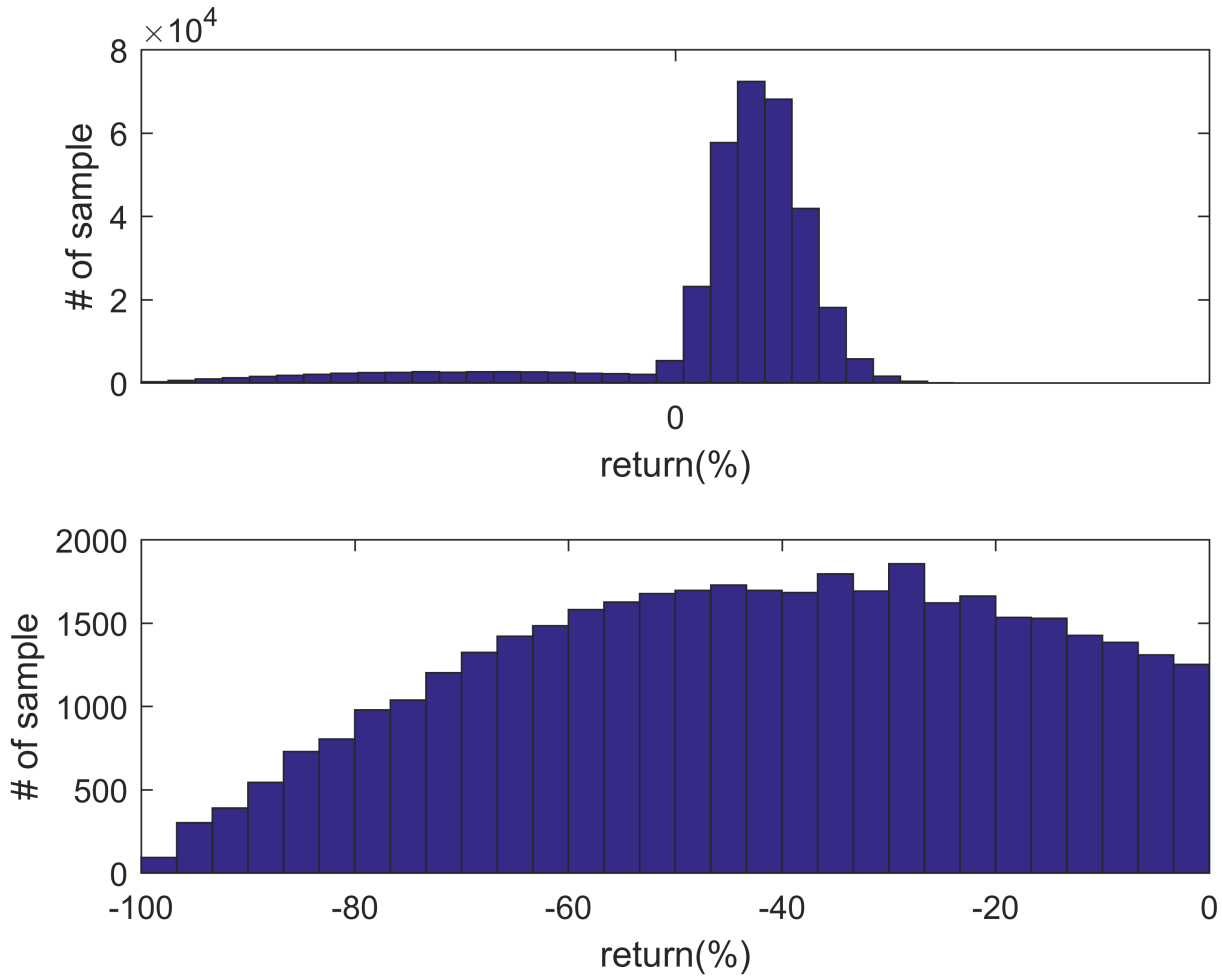

Figure 5.1: Histogram of investment return in LC.

5.2 Investment decision model

The investment decision model of this study is a series of two processes: classification and regression. First, imbalanced classification determines the decision surface that separates majority data from minority data when data is imbalanced. In this case, since we assign a greater importance to the minority data, based on the decision surface, there are very few minority data on one side and there are a lot on the other side. On the other hand, majority data is widely distributed regardless of the decision surface because of its relatively small importance. Applying this to P2P data, fully paid loans are widespread, while default loans are separated by decision surface. Therefore, the first step is to discard the loans that are expected to be default through the proposed IEFSVM and select the remaining loans.

Secondly, Serrano-Cinca and Gutierrez-Nieto [31] demonstrate that a portfolio with loans that are predicted to achieve a high return using simple regression method can be profitable. Table 5.2 demonstrates the difference between credit scoring and profit scoring. Therefore, the next step is to construct a multiple linear regression for selected loans through IEFSVM. In this case, independent variables are the same as those used in IEFSVM. If the regression equation is established, we can predict the investment return of the test set's loans. We choose the loans with the investment return of the top 10% in the

Table 5.2: Comparison of credit scoring and profit scoring

Credit scoring	Loan Status(Y) $=$ $\begin{cases} 0 & \text{if default} \\ 1 & \text{if fully paid} \end{cases}$, $Y = f(X_1, ..., X_n)$
Profit scoring	$IRR = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_n X_n$

test set loans. The final investment model is to invest the same amount of money on selected loans, and the process of the proposed investment decision model is shown in Figure 5.2.

To be specific, we first randomly extract 1000 samples from raw data and split them by 60 to 40. Then, the three steps are composed of data preprocessing, classification modeling, and regression modeling. After the steps, we can obtain the final return by constructing a portfolio with top decile of predicted return.

From a total of 331,878 samples, 1000 samples are randomly extracted for one experiment, and IR changes slightly at each extraction. IR for all data is 6.6329, and when 1000 samples are extracted, it spreads with standard deviation of 0.6291 around 6.6329.

The investment decision model can provide the performance of IEFSVM for both classification and profitability, and this is the reason for separating IEFSVM and regression. If IEFSVM is effective, the proposed portfolio using IEFSVM could demonstrate a higher investment return than that of Serrano-

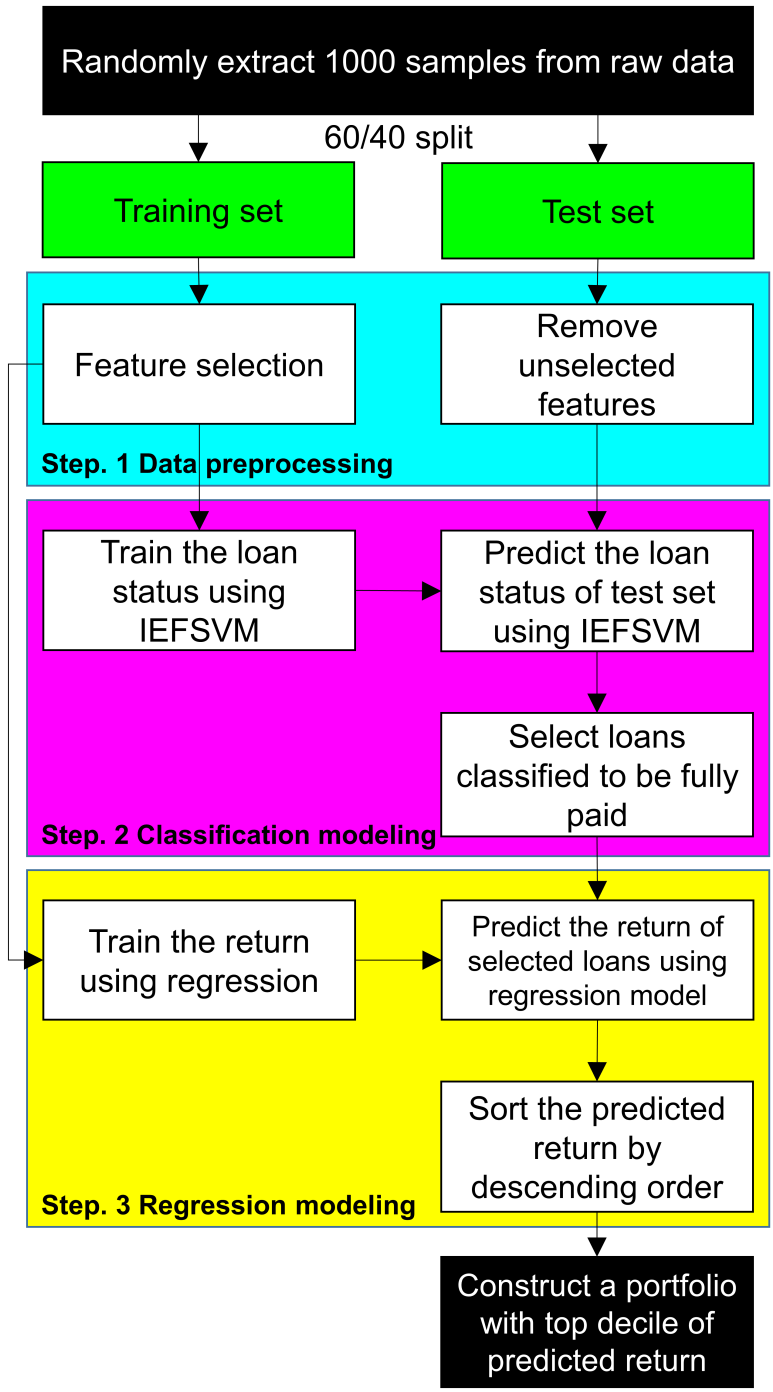

Figure 5.2: Investment decision model.

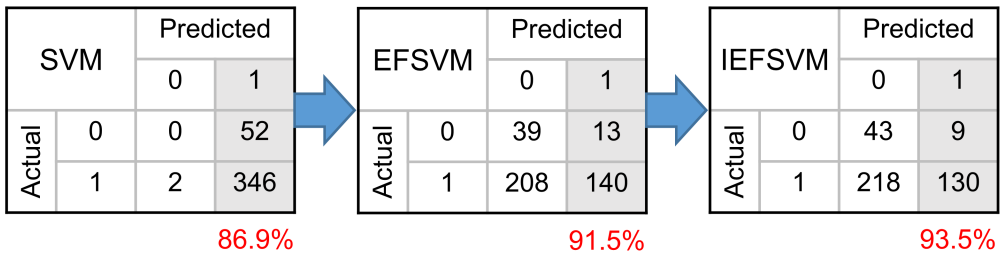

Figure 5.3: Confusion matrix example according to classifier.

Table 5.3: Classification metrics of the example according to classifier

	SVM	EFSVM	IEFSVM
Precision	$\frac{346}{346+52} = 86.9\%$	$\frac{140}{140+13} = 91.5\%$	$\frac{130}{130+9} = 93.5\%$
Predicted negative condition rate	$1 - \frac{346+52}{400} = 0.5\%$	$1 - \frac{140+13}{400} = 61.8\%$	$1 - \frac{130+9}{400} = 65.3\%$

Cinca and Gutierrez-Nieto [31].

For example, Figure 5.3 is a confusion matrix based on the prediction results of 400 test samples with SVM, EFSVM, and IEFSVM. Table 5.3 specifies precision and predicted negative condition rate of the example. Then, precision vales of SVM, EFSVM, and IEFSVM are 87.4%, 91.5% and 93.5%, respectively, which indicates that IEFSVM is effective. However, the disadvantage of the investment decision is that the higher the performance of classifier, the more data it discards. For SVM, it filters 0.5% of data, but with EFSVM, 61.8% and IEFSVM, 65.3%.

Alternatively, the proposed investment decision model is similar to end-to-end learning algorithms such as Autoencoder or Learning-to-rank, however, there is a definite difference between the two. First, Autoencoder has an encoding and a decoding process. There is a hidden layer between input and output, then data compression occurs in the process from input to hidden layer, which is called encoding. In the process of going from the hidden layer to the output, the process of extracting the compressed data occurs, which is called decoding. Thus, it can be misunderstood that the process of predicting loans through the IEFSVM is the encoding process, whereas the process of selecting loans through the regression is a decoding process. However, the classification and regression training procedures in our model are independent. The response variable is different from the loan status and investment return although it performs classification and regression using the same explanatory variables. Therefore, our proposed algorithm is a combination of two independent processes. On the other hand, Autoencoder is different from our algorithm because encoding and decoding processes are not independent and decoding occurs through encoded data.

Secondly, Learning-to-rank is a learning algorithm which trains the model for ranking task. In our model, we train the investment return of training data through regression and predict the return of test data. Then, the proposed model also selects loans with the top 10% return to construct a portfolio. This

process is a concept similar to Learning-to-rank. However, our proposed model additionally removes the loans predicted to be default. That is, instead of setting the rank of all the loans, our model only considers the loans predicted to be fully paid. On the other hand, Learning-to-rank algorithm sets a ranking for all test sets, which is a major difference between Learning-to-rank and our model.

5.3 Empirical study

In this section, we first demonstrate the benchmark algorithms and the parameter settings of imbalanced classification. Also, performance metrics for imbalanced classification and investment decision are defined. Then, we compare our proposed model with other imbalanced classification models and other investment decision models.

5.3.1 Benchmark algorithms and parameter settings

As in Section 4.2, seven benchmark algorithms are compared to evaluate the classification performance of IEFSVM. The algorithms are comprised of cs-AdaBoost [66], cs-RF [67], cs-XGBoost [34], EasyEnsemble [68], RUSBoost [69], w-ELM [70], and EFSVM.

The radial basis function (RBF) kernel or linear kernel is used, and the regularization parameter C is selected from $\{2^{-6}, 2^{-4}, ..., 2^4, 2^6\}$ for SVM-based learning machines such as EFSVM and IEFSVM. The neighborhood size is chosen from $\{1, 3, 5, 7, 9, 11, 13, 15\}$ to calculate the nearest neighbors entropy. We choose 100 maximum learning iterations for tree-based learning machines such as cs-AdaBoost, cs-RF, Easyensemble, and RUSBoost, while the tuning of cs-XGBoost conforms to the guideline of Xia *et al.* [34] and Jain [99]. We tune the parameters of each classifier by a 5-fold cross-validation procedure.

5.3.2 Performance metrics

To adequately assess the performance of our proposed investment decision model, we employ three classification metrics, and two profitability metrics. The classification performance metrics consist of AUC, precision, and predicted negative condition rate. To denote these metrics, we first define a confusion matrix that describes the performance of classification model. Also, to apply P2P data, classes of the confusion matrix consist of fully paid and default.

Table 5.4: Confusion matrix

Total population		Predicted	
		Fully paid	Default
Actual	Fully paid	True Fully paid	False Default
	Default	False Fully paid	True Default

AUC [11, 76] is used to determine which of the classification models predict the classes best, and AUC in the binary classification is defined as follows.

$$\text{AUC} \;=\; (1 + TP - FP) \;/\; 2. \tag{5.1}$$

where TP and FP denote the ratio of actual positives correctly classified and that of the negatives misclassified, respectively.

In our proposed investment decision model, there is a process which discards the loans predicted to be default using IEFSVM. Then, we can define the ratio of discarding samples to whole samples as follows.

$$\text{Predicted negative condition rate} = \text{Predicted Default} \;/\; \text{Total Population}. \quad (5.2)$$

After discarding the loans predicted to be default, we can measure the ratio of fully paid loans for the remaining samples as follows.

$$\text{Precision} = \text{True Fully paid} \;/\; \text{Predicted Fully paid}. \quad (5.3)$$

The profitability performance metrics are comprised of investment return, and Sharpe ratio of each portfolio.

5.3.3 Comparison of classifiers

The proposed investment decision model can use a classifier other than IEFSVM in the process of discarding loans predicted to be default. Then, to compare the performance of IEFSVM, we calculate the results of seven state-of-the-art classifiers specified in Section 5.3.1. Performance metrics consist of the three imbalanced classification metrics and two profitability metrics as described above, and the mean and standard deviation of total 1000 results are specified in Table 5.5. Also, we rank each metric below each value. To verify whether the proposed IEFSVM significantly surpasses the benchmarks, Table 5.6 performs hypothesis tests with a significance level of 0.05.

On the whole, results in Table 5.5 show that IEFSVM is superior to other algorithms. For AUC value, IEFSVM performs the best, however, it does not outperform cs-AdaBoost. Table 5.6 demonstrates that IEFSVM does not statistically surpass cs-AdaBoost. Meanwhile, for precision and predicted negative condition rate, IEFSVM significantly outperforms other algorithms. With regard to profitability metrics, IEFSVM statistically outperforms other algorithms for both investment return and Sharpe ratio. Specifically, cs-AdaBoost and EasyEnsemble demonstrate an unusual pattern. That is, cs-AdaBoost shows high performance on imbalanced classification metrics, whereas it shows below par performance on profitability metrics. Conversely, EasyEnsemble in-

dicates high performance on profitability, but low performance on classification. This suggests the performance can vary according to the metrics, however, IEFSVM shows a robust performance. In addition, IEFSVM generally beats EFSVM, which supports our proposed model to modify EFSVM as an instance-based model.

Table 5.5: Comparison of classifiers

	cs-AdaBoost	cs-RF	EasyEnsemble	RUSBoost	w-ELM	cs-XGBoost	EFSVM	IEFSVM
AUC	59.23 ± 3.95	52.58 ± 2.63	55.28 ± 3.95	58.37 ± 5.07	56.52 ± 3.64	56.62 ± 4.15	57.14 ± 3.85	59.38 ± 2.85
	2	8	7	3	6	5	4	1
Precision	91.98 ± 2.59	87.63 ± 0.65	89.20 ± 1.65	91.39 ± 2.81	89.47 ± 1.42	89.26 ± 1.48	90.42 ± 2.13	92.16 ± 1.83
	2	8	7	3	5	6	4	1
Predicted negative condition rate	52.04 ± 19.26	7.49 ± 1.84	45.37 ± 5.36	50.24 ± 20.12	39.45 ± 6.18	31.73 ± 6.93	47.61 ± 13.85	57.28 ± 8.85
	2	8	5	3	6	7	4	1
Return with top 10%	10.06 ± 4.63	12.26 ± 5.51	11.91 ± 4.93	10.34 ± 4.48	11.44 ± 5.34	11.58 ± 5.00	12.63 ± 5.59	14.99 ± 2.87
	8	3	4	7	6	5	2	1
Return / Standard dev.	2.175	2.226	2.417	2.31	2.145	2.319	2.26	5.227
	7	6	2	4	8	3	5	1

Table 5.6: Significance tests of classifiers

	cs-AdaBoost	cs-RF	EasyEnsemble	RUSBoost	w-ELM	cs-XGBoost	EFSVM
AUC	Not rejected	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**
	0.1651	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
Precision	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**
	0.0364	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
Predicted negative condition rate	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**
	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
Return with top 10%	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Rejected**
	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

5.3.4 Comparison of investment decision model

There are two recent studies of investment decision model in P2P lending. Serrano-Cinca and Gutierrez-Nieto [31] suggested a profit scoring scheme for decision support system. They employed the internal rate of return (IRR) which is used to estimate the profitability of potential investments. Let this method be Benchmark 1. In the meantime, Guo *et al.* [32] proposed an instance-based credit risk evaluation model. After quantifying the loan's return and risk, they solved a portfolio optimization problem to develop the investment decision model. Let this method be Benchmark 2. Table 5.7 shows the profitability performance of IEFSVM and two benchmarks. The metrics consist of investment return, standard deviation, and Sharpe ratio of each model.

On the whole, results in Table 5.7 show that IEFSVM surpasses other investment decision models. Benchmark 1 has a higher return than Benchmark 2, whereas Benchmark 2 is better in terms of the Sharpe ratio. IEFSVM shows a higher standard deviation than Benchmark 2, however, it outperforms both benchmarks with respect to return and Sharpe ratio. Therefore, our proposed model substantially improves the profitability performances compared with existing models.

Table 5.7: Comparison of the investment decision model

	Benchmark1	Benchmark2	IEFSVM
Return(%)	12.475	8.707	14.99
Standard dev.(%)	5.687	2.254	2.868
Return / Standard dev.	2.193	3.863	5.227

Overall, by experimental studies, the performance of our model is verified with two types of benchmarks, that is, the seven imbalanced classifiers and two investment decision models. For imbalanced classifiers, IEFSVM significantly surpasses other classifiers in terms of AUC, precision, predicted negative condition rate, returns with top 10%, and Sharpe ratio only except cs-AdaBoost in terms of AUC. Since the loan status prediction problem of P2P lending market distinguishes the minority data from the majority data, the application of IEFSVM successfully improves the classification performances and well predicts fully paid loans. In case of P2P lending data, the IR is very high as 6.63, and since many majority data exceed the decision surface, the entropy value is highly varied. As a result, the proposed algorithm with the combination of entropy could better reflect the information and lead to high classification performance. Also, for investment decision models, the proposed model statistically surpasses the existing investment decision models in terms of investment return and Sharpe ratio. Thus, based on empirical study, IEFSVM is a decent classification model for investment decision in P2P lending market, and it can be employed in other areas as well.

Chapter 6

Conclusion

6.1 Contributions

This dissertation presents a new classifier that better predicts the class imbalance problem. First, by identifying the characteristics of nearest neighbors entropy through the graphical analysis, we present a way to better quantify the uncertainty of information. Then, instance-based entropy fuzzy support vector machine (IEFSVM) is introduced to better classify binary imbalanced datasets. It transforms the instance-based entropy into polar coordinate to develop the entropy appropriate for each sample. Considering that the existing EFSVM employs a unified neighborhood size when determining fuzzy membership, the proposed algorithm combine nearest neighbors entropies by neighborhood size for each data point. Then, the classifier can reflect all information in entropy of each instance efficiently. Furthermore, the graphical analysis of nearest neighbors entropy not only demonstrates the pattern of the entropy, but also indicates rational reasoning to the fuzzy membership. In ad-

dition, The two engineering theories, information theory and data mining, are incorporated in an industrial engineering way to create a model that can predict social phenomena in more detail. In experimental studies, the classification performance of IEFSVM is superior to that of other benchmarks including the existing EFSVM for both public and real-world imbalanced datasets, making the most of entropy as a new way. In particular, IEFSVM has a strong advantage in classifying imbalanced datasets because the importance of majority data is low when dealing with datasets with high imbalance ratio. The majority data is widely distributed regardless of the minority data, and the nearest neighbors entropy for majority data can highly vary depending on the neighborhood size. The proposed IEFSVM better reflects this information, which leads to better classification of imbalanced datasets. Through the analysis, we have obtained a deeper understanding of nearest neighbors entropy and proposed a solution to the problem of unified neighborhood size for all data points which has been a long-standing problem in data mining. The class imbalance problem is a widespread problem in other fields, in this dissertation, we applied the proposed classifier in P2P market to investigate whether it still has a robust performance of imbalanced classification.

Several financial crises have stalled the growth of financial market, and interest rate has also decreased, resulting in lower profits of bank lending and brokerage. In addition, due to the distrust of consumers in the existing financial

sector and the rapid development of IT technology, global IT companies utilizing FinTech have entered the financial market. Therefore, this dissertation will satisfy investors' needs for a new profit model by studying the investment decision in the P2P lending market. To achieve the purpose, this dissertation assumes the loan status prediction problem in P2P lending market as a class imbalance problem. This provides a new perspective on the data. Finally, using the novel cost-sensitive classifier, we develop an investment decision to obtain high profits in P2P lending market. Specifically, filtering through IEFSVM and ranking the loans through regression in investment decision is a new direction of data observation that can be applied to other areas as well. This is also a contribution to enhance the investment decision proposed by Serrano-Cinca and Gutierrez-Nieto [31] via choosing loans predicted as fully paid by IEFSVM. The analysis of financial market with an industrial engineering perspective contributes not only to the financial market but also to the engineering methodology. The investment decision of P2P lending market can be regarded as limited research, however, the class imbalance problem is widespread in the real world. Therefore, we can make better prediction through the classifier developed in this dissertation, and present an application method of filtering the minority data, contributing much to other fields.

6.2 Future Work

This dissertation also has a direction of development that should be handled in future work. First, only eight neighborhood sizes are considered when constructing the entropy pairs. Instead, we can generally analyze how the pattern of nearest neighbors entropy changes depending on how many neighborhood sizes are assigned. Since the entropy differs according to neighborhood size in spite of fixed data points, the analysis of varying neighborhood size is expected to better understand nearest neighbors entropy. Secondly, the proposed fuzzy membership only employs the mean and standard deviation of entropy pairs. Of course, this dissertation derives the rational fuzzy membership through the graphical pattern of entropy pairs, however, more elaborate instance-based entropy fuzzy membership can be developed not necessarily with the polar coordinates. Lastly, this investment decision model increases the discarded amount of data as the use of classifiers with high classification performance. For general SVM filtering 42.9% of the data, but using the proposed classifier discards 59.1%. As a result, if the investment amount is extremely large, it will not detect any more investment opportunities. Instead, we need to measure how the investment return and Sharpe ratio change by the investment amount. Also, constructing more sophisticated model rather than simple multivariate regression when ranking the loans will enhance the profitability of portfolio.

Bibliography

[1] A. FernáNdez, V. LóPez, M. Galar, M. J. Del Jesus, and F. Herrera, "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches," *Knowledge-based systems*, vol. 42, pp. 97–110, 2013.

[2] Q. Yang and X. Wu, "10 challenging problems in data mining research," *International Journal of Information Technology & Decision Making*, vol. 5, no. 04, pp. 597–604, 2006.

[3] J. Tian, H. Gu, and W. Liu, "Imbalanced classification using support vector machine ensemble," *Neural Computing and Applications*, vol. 20, no. 2, pp. 203–209, 2011.

[4] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[5] K. Yoon and S. Kwek, "A data reduction approach for resolving the imbalanced data issue in functional genomics," *Neural Computing and*

Applications, vol. 16, no. 3, pp. 295–306, 2007.

[6] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.

[7] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE transactions on neural networks*, vol. 13, no. 2, pp. 464–471, 2002.

[8] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[9] Y. Chen, K. Wu, X. Chen, C. Tang, and Q. Zhu, "An entropy-based uncertainty measurement approach in neighborhood systems," *Information Sciences*, vol. 279, pp. 239–250, 2014.

[10] Q. Fan, Z. Wang, D. Li, D. Gao, and H. Zha, "Entropy-based fuzzy support vector machine for imbalanced datasets," *Knowledge-Based Systems*, vol. 115, pp. 87–99, 2017.

[11] J. Huang and C. X. Ling, "Using auc and accuracy in evaluating learning algorithms," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.

[12] Z. Pan, Y. Wang, and W. Ku, "A new k-harmonic nearest neighbor classifier based on the multi-local means," *Expert Systems with Applications*, vol. 67, pp. 115–125, 2017.

[13] J. Gou, Y. Zhan, Y. Rao, X. Shen, X. Wang, and W. He, "Improved pseudo nearest neighbor classification," *Knowledge-Based Systems*, vol. 70, pp. 361–375, 2014.

[14] Ö. F. Ertuğrul and M. E. Tağluk, "A novel version of k nearest neighbor: Dependent nearest neighbor," *Applied Soft Computing*, vol. 55, pp. 480–490, 2017.

[15] Y. Zhu, Z. Wang, and D. Gao, "Gravitational fixed radius nearest neighbor for imbalanced problem," *Knowledge-Based Systems*, vol. 90, pp. 224–238, 2015.

[16] X. Zhang, Y. Li, R. Kotagiri, L. Wu, Z. Tari, and M. Cheriet, "Krnn: k rare-class nearest neighbour classification," *Pattern Recognition*, vol. 62, pp. 33–44, 2017.

[17] F. Bulut and M. F. Amasyali, "Locally adaptive k parameter selection for nearest neighbor classifier: one nearest cluster," *Pattern Analysis and Applications*, vol. 20, no. 2, pp. 415–425, 2017.

[18] P. Saha, I. Bose, and A. Mahanti, "A knowledge based scheme for risk assessment in loan processing by banks," *Decision Support Systems*, vol. 84, pp. 78–88, 2016.

[19] R. Tsaih, Y.-J. Liu, W. Liu, and Y.-L. Lien, "Credit scoring system for small business loans," *Decision Support Systems*, vol. 38, no. 1, pp. 91–99, 2004.

[20] L. Puro, J. E. Teich, H. Wallenius, and J. Wallenius, "Borrower decision aid for people-to-people lending," *Decision Support Systems*, vol. 49, no. 1, pp. 52–60, 2010.

[21] A. Marques, V. García, and J. S. Sánchez, "A literature review on the application of evolutionary computing to credit scoring," *Journal of the Operational Research Society*, vol. 64, no. 9, pp. 1384–1399, 2013.

[22] R. T. Stewart, "A profit-based scoring system in consumer credit: making acquisition decisions for credit cards," *Journal of the Operational Research Society*, vol. 62, no. 9, pp. 1719–1725, 2011.

[23] T. Verbraken, C. Bravo, R. Weber, and B. Baesens, "Development and application of consumer credit scoring models using profit-based classification measures," *European Journal of Operational Research*, vol. 238, no. 2, pp. 505–513, 2014.

[24] S. Maldonado, C. Bravo, J. Lopez, and J. Perez, "Integrated framework for profit-based feature selection and svm classification in credit scoring," *Decision Support Systems*, vol. 104, pp. 113–121, 2017.

[25] F. Zhu, J. Yang, C. Gao, S. Xu, N. Ye, and T. Yin, "A weighted one-class support vector machine," *Neurocomputing*, vol. 189, pp. 1–10, 2016.

[26] F. Zhu, J. Yang, N. Ye, C. Gao, G. Li, and T. Yin, "Neighbors' distribution property and sample reduction for support vector machines," *Applied Soft Computing*, vol. 16, pp. 201–209, 2014.

[27] F. Zhu, N. Ye, W. Yu, S. Xu, and G. Li, "Boundary detection and sample reduction for one-class support vector machines," *Neurocomputing*, vol. 123, pp. 166–173, 2014.

[28] F. Zhu, J. Yang, J. Gao, and C. Xu, "Extended nearest neighbor chain induced instance-weights for svms," *Pattern Recognition*, vol. 60, pp. 863–874, 2016.

[29] Y. Chen and Y. Hao, "A feature weighted support vector machine and k-nearest neighbor algorithm for stock market indices prediction," *Expert Systems with Applications*, vol. 80, pp. 340–355, 2017.

[30] P. Cho, M. Lee, and W. Chang, "Instance-based entropy fuzzy support vector machine for imbalanced data," *arXiv preprint arXiv:1807.03933*, 2018.

[31] C. Serrano-Cinca and B. Gutiérrez-Nieto, "The use of profit scoring as an alternative to credit scoring systems in peer-to-peer (p2p) lending," *Decision Support Systems*, vol. 89, pp. 113–122, 2016.

[32] Y. Guo, W. Zhou, C. Luo, C. Liu, and H. Xiong, "Instance-based credit risk assessment for investment decisions in p2p lending," *European Journal of Operational Research*, vol. 249, no. 2, pp. 417–426, 2016.

[33] W. F. Sharpe, "The sharpe ratio," *Journal of portfolio management*, vol. 21, no. 1, pp. 49–58, 1994.

[34] Y. Xia, C. Liu, and N. Liu, "Cost-sensitive boosted tree for loan evaluation in peer-to-peer lending," *Electronic Commerce Research and Applications*, vol. 24, pp. 30–49, 2017.

[35] Y. Xu, J. Yu, and Y. Zhang, "Knn-based weighted rough ν-twin support vector machine," *Knowledge-Based Systems*, vol. 71, pp. 303–313, 2014.

[36] X. Pan, Y. Luo, and Y. Xu, "K-nearest neighbor based structural twin support vector machine," *Knowledge-Based Systems*, vol. 88, pp. 34–44, 2015.

[37] Z. Li, J. Tang, and F. Guo, "Learning from real imbalanced data of 14-3-3 proteins binding specificity," *Neurocomputing*, vol. 217, pp. 83–91, 2016.

[38] H. Yu, J. Ni, and J. Zhao, "Acosampling: An ant colony optimization-based undersampling method for classifying imbalanced dna microarray data," *Neurocomputing*, vol. 101, pp. 309–318, 2013.

[39] A. Mellor, S. Boukir, A. Haywood, and S. Jones, "Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 155–168, 2015.

[40] E. A. Freeman, G. G. Moisen, and T. S. Frescino, "Evaluating effectiveness of down-sampling for stratified designs and unbalanced prevalence in random forest models of tree species distributions in nevada," *Ecological modelling*, vol. 233, pp. 1–10, 2012.

[41] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3446–3453, 2012.

[42] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with dte-sbd: Decision tree ensemble based on smote and bagging with differentiated sampling rates," *Information Sciences*, vol. 425, pp. 76–91, 2018.

[43] B. Zhu, Y. Niu, J. Xiao, and B. Baesens, "A new transferred feature selection algorithm for customer identification," *Neural Computing and Applications*, vol. 28, no. 9, pp. 2593–2603, 2017.

[44] A. Idris, A. Khan, and Y. S. Lee, "Intelligent churn prediction in telecom: employing mrmr feature selection and rotboost based ensemble classification," *Applied intelligence*, vol. 39, no. 3, pp. 659–672, 2013.

[45] K. Polat, "Similarity-based attribute weighting methods via clustering algorithms in the classification of imbalanced medical datasets," *Neural Computing and Applications*, vol. 30, no. 3, pp. 987–1013, 2018.

[46] A. Artetxe, M. Graña, A. Beristain, and S. Ríos, "Balanced training of a hybrid ensemble method for imbalanced datasets: a case of emergency department readmission prediction," *Neural Computing and Applications*, pp. 1–10, 2017.

[47] Y. Huang and T. Kechadi, "An effective hybrid learning system for telecommunication churn prediction," *Expert Systems with Applications*,

vol. 40, no. 14, pp. 5635–5647, 2013.

[48] B. M. Abidine, L. Fergani, B. Fergani, and M. Oussalah, "The joint use of sequence features combination and modified weighted svm for improving daily activity recognition," *Pattern Analysis and Applications*, vol. 21, no. 1, pp. 119–138, 2018.

[49] S. Liu, Y. Wang, J. Zhang, C. Chen, and Y. Xiang, "Addressing the class imbalance problem in twitter spam detection using ensemble learning," *Computers & Security*, vol. 69, pp. 35–49, 2017.

[50] E. Radkani, S. Hashemi, A. Keshavarz-Haddad, and M. A. Haeri, "An entropy-based distance measure for analyzing and detecting metamorphic malware," *Applied Intelligence*, pp. 1–11, 2018.

[51] X.-Y. Lu, M.-S. Chen, J.-L. Wu, P.-C. Chang, and M.-H. Chen, "A novel ensemble decision tree based on under-sampling and clonal selection for web spam detection," *Pattern Analysis and Applications*, pp. 1–14, 2017.

[52] R. Batuwita and V. Palade, "Fsvm-cil: fuzzy support vector machines for class imbalance learning," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 558–571, 2010.

[53] X. Jiang, Z. Yi, and J. C. Lv, "Fuzzy svm with a new fuzzy membership function," *Neural Computing & Applications*, vol. 15, no. 3-4, pp. 268–276, 2006.

[54] H.-L. Dai, "Class imbalance learning via a fuzzy total margin based support vector machine," *Applied Soft Computing*, vol. 31, pp. 172–184, 2015.

[55] J. P. Hwang, S. Park, and E. Kim, "A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost function," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8580–8585, 2011.

[56] S. Ando, "Classifying imbalanced data in distance-based feature space," *Knowledge and Information Systems*, vol. 46, no. 3, pp. 707–730, 2016.

[57] Z. Chen, T. Lin, R. Chen, Y. Xie, and H. Xu, "Creating diversity in ensembles using synthetic neighborhoods of training samples," *Applied Intelligence*, vol. 47, no. 2, pp. 570–583, 2017.

[58] Z. Chen, T. Lin, X. Xia, H. Xu, and S. Ding, "A synthetic neighborhood generation based ensemble learning for the imbalanced data classification," *Applied Intelligence*, vol. 48, no. 8, pp. 2441–2457, 2018.

[59] C. Kaleli, "An entropy-based neighbor selection approach for collaborative filtering," *Knowledge-Based Systems*, vol. 56, pp. 273–280, 2014.

[60] T. Zheng and L. Zhu, "Uncertainty measures of neighborhood system-based rough sets," *Knowledge-Based Systems*, vol. 86, pp. 57–65, 2015.

[61] Y. Chen, Y. Xue, Y. Ma, and F. Xu, "Measures of uncertainty for neighborhood rough sets," *Knowledge-Based Systems*, vol. 120, pp. 226–235, 2017.

[62] C. Zhu and Z. Wang, "Entropy-based matrix learning machine for imbalanced data sets," *Pattern Recognition Letters*, vol. 88, pp. 72–80, 2017.

[63] D. Gupta, B. Richhariya, and P. Borah, "A fuzzy twin support vector machine based on information entropy for class imbalance learning," *Neural Computing and Applications*, pp. 1–12.

[64] D. Gupta and B. Richhariya, "Entropy based fuzzy least squares twin support vector machine for class imbalance learning," *Applied Intelligence*, pp. 1–20, 2018.

[65] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[66] Y. Freund, R. E. Schapire, *et al.*, "Experiments with a new boosting algorithm," in *Icml*, vol. 96, pp. 148–156, Citeseer, 1996.

[67] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[68] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2009.

[69] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[70] W. Zong, G.-B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.

[71] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A novel ensemble method for classifying imbalanced data," *Pattern Recognition*, vol. 48, no. 5, pp. 1623–1637, 2015.

[72] L. Nanni, C. Fantozzi, and N. Lazzarini, "Coupling different methods for overcoming the class imbalance problem," *Neurocomputing*, vol. 158, pp. 48–61, 2015.

[73] B. Zhu, B. Baesens, and S. K. vanden Broucke, "An empirical comparison of techniques for the class imbalance problem in churn prediction," *Information sciences*, vol. 408, pp. 84–99, 2017.

[74] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," *IEEE transactions on cybernetics*, vol. 47, no. 12, pp. 4263–4274, 2017.

[75] K. Li, X. Kong, Z. Lu, L. Wenyin, and J. Yin, "Boosting weighted elm for imbalanced learning," *Neurocomputing*, vol. 128, pp. 15–21, 2014.

[76] T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," *Machine learning*, vol. 31, no. 1, pp. 1–38, 2004.

[77] C. Serrano-Cinca, B. Gutierrez-Nieto, and L. López-Palacios, "Determinants of default in p2p lending," *PloS one*, vol. 10, no. 10, p. e0139427, 2015.

[78] C. Jiang, Z. Wang, R. Wang, and Y. Ding, "Loan default prediction by combining soft information extracted from descriptive text in online peer-to-peer lending," *Annals of Operations Research*, vol. 266, no. 1-2, pp. 511–529, 2018.

[79] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of

research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124–136, 2015.

[80] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring," *Journal of the operational research society*, vol. 54, no. 6, pp. 627–635, 2003.

[81] G. G. Chen and T. Åstebro, "Bound and collapse bayesian reject inference for credit scoring," *Journal of the Operational Research Society*, vol. 63, no. 10, pp. 1374–1387, 2012.

[82] K. Kennedy, B. M. Namee, and S. J. Delany, "Using semi-supervised classifiers for credit scoring," *Journal of the Operational Research Society*, vol. 64, no. 4, pp. 513–529, 2013.

[83] J. Ouenniche, K. Bouslah, J. M. Cabello, and F. Ruiz, "A new classifier based on the reference point method with application in bankruptcy prediction," *Journal of the Operational Research Society*, 2018.

[84] C. Liberati and F. Camillo, "Personal values and credit scoring: new insights in the financial prediction," *Journal of the Operational Research Society*, pp. 1–12, 2018.

[85] A. A. Aduenko, A. P. Motrenko, and V. V. Strijov, "Object selection in credit scoring using covariance matrix of parameters estimations," *Annals of Operations Research*, vol. 260, no. 1-2, pp. 3–21, 2018.

[86] Z. Affes and R. Hentati-Kaffel, "Forecast bankruptcy using a blend of clustering and mars model: case of us banks," *Annals of Operations Research*, pp. 1–38, 2016.

[87] A. I. Marqués, V. García, and J. S. Sánchez, "On the suitability of resampling techniques for the class imbalance problem in credit scoring," *Journal of the Operational Research Society*, vol. 64, no. 7, pp. 1060–1070, 2013.

[88] J. Sun, Z. Shang, and H. Li, "Imbalance-oriented svm methods for financial distress prediction: a comparative study among the new sb-svm-ensemble method and traditional methods," *Journal of the Operational Research Society*, vol. 65, no. 12, pp. 1905–1919, 2014.

[89] W. Li, S. Ding, Y. Chen, and S. Yang, "Heterogeneous ensemble for default prediction of peer-to-peer lending in china," *IEEE Access*, 2018.

[90] Y. Xia, X. Yang, and Y. Zhang, "A rejection inference technique based on contrastive pessimistic likelihood estimation for p2p lending," *Electronic Commerce Research and Applications*, 2018.

[91] Y. Xia, C. Liu, Y. Li, and N. Liu, "A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring," *Expert Systems with Applications*, vol. 78, pp. 225–241, 2017.

[92] X. Zeng, L. Liu, S. Leung, J. Du, X. Wang, and T. Li, "A decision support model for investment on p2p lending platform," *PloS one*, vol. 12, no. 9, p. e0184242, 2017.

[93] A. Lemmens and C. Croux, "Bagging and boosting classification trees to predict churn," *Journal of Marketing Research*, vol. 43, no. 2, pp. 276–286, 2006.

[94] B. Zhu, B. Baesens, A. Backiel, and S. K. Vanden Broucke, "Benchmarking sampling techniques for imbalance learning in churn prediction," *Journal of the Operational Research Society*, vol. 69, no. 1, pp. 49–65, 2018.

[95] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[96] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[97] E. Osuna and F. Girosi, "Reducing the run-time complexity of support vector machines," in *International Conference on Pattern Recognition (submitted)*, 1998.

[98] P. Murphy, "Uci-benchmark repository of artificial and real data sets," *University of California Irvine*, 1995.

[99] A. Jain, "Complete guide to parameter tuning in xgboost," 2016.

[100] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[101] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics*, pp. 65–70, 1979.

[102] C. Beyan and R. Fisher, "Classifying imbalanced data sets using similarity based hierarchical decomposition," *Pattern Recognition*, vol. 48, no. 5, pp. 1653–1672, 2015.

[103] National cancer institute, "Aids antiviral screen data - nci dtp data - national cancer institute - confluence wiki," May 2004.

[104] Gokagglers, "Cervical cancer risk classification," Aug 2017.

[105] Lending Club, "Lending club statistics," Jan 2016.

[106] Otto group, "Otto group product classification," May 2015.

[107] Seoul Metropolitan Government, "Daily weather of seoul," May 2018.

[108] Seoul Metropolitan Government, "Daily air pollution of seoul," May 2018.

[109] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

국문초록

클래스 불균형 데이터를 바탕으로 한 지도학습은 많은 분야에서 중요한 문제로 여겨져 왔다. 소수 데이터의 무시로 인해 일반적인 분류 알고리즘과는 다른 방법이 필요하기 때문이다. 이러한 맥락에서, 퍼지 서포트 벡터 머신(Fuzzy Support Vector Machine, FSVM)은 클래스 불균형 데이터를 처리하기 위해 각 데이터 포인트의 가중치를 다르게 할당할 수 있으며, 가중치를 결정하는 연구들이 활발하게 수행되었다. 그러한 방법들 중에서 정보 이론의 엔트로피는 데이터의 설명력을 가지고 있기 때문에 퍼지 서포트 벡터 머신에 적용 할 수 있다. 또한, 클래스 불균형 분류에 대한 정보의 확실성을 정량화하기 위해 최근점 이웃의 클래스에 기반한 최근접 이웃 엔트로피 개념이 제안되었다. 그러나 기존의 엔트로피 퍼지 서포트 벡터 머신(Entropy Fuzzy Support Vector Machine, EFSVM)은 모델을 학습할 때 통일된 이웃 크기를 사용하여 오분류를 유발한다. 그래서 이 논문은 이웃의 클래스를 보다 잘 반영하는 새로운 사례 기반 분류기를 개발하는 것을 목표로 한다. 먼저, 제안된 사례 기반 엔트로피 퍼지 서포트 벡터 머신(Instance-based Entropy Fuzzy Support Vector Machine, IEFSVM)은 최근접 이웃 엔트로피의 그래프 패턴을 기반으로 개발되었다. 고정된 데이터 포인트에 대해 엔트로피 값이 이웃 크기에 따라 달라질 수 있다는 것을 참고한다면, 여러 이웃 크기에 따른 엔트로피 조합을 고려할 수 있다. 그리고 그 엔트로피 조합의 그래픽 패턴을 사용하여 합리적인 추론을 통해 가중치를 할당한다. 두 번째로,

공공 데이터와 실제 데이터를 사용하여 여러 벤치마크를 통해 IEFSVM의 성능을 입증한다. IEFSVM의 기본 분류기는 서포트 벡터 머신(Support Vector Machine, SVM)이기 때문에, 벤치마크를 구성할 때 SVM을 기본 분류기로 사용하는 알고리즘과 그렇지 않은 알고리즘 두 가지를 사용한다. 특히, 제안된 IEFSVM은 EFSVM을 포함한 다른 벤치마크들보다 높은 수신자 조작 특성 곡선의 밑 면적(Area Under the receiver operating characteristic Curve, AUC) 값을 가지며 통계적으로 개선된 예측 성능을 보여준다. 마지막으로 Peer-to-peer(P2P) 대출 시장에 IEFSVM 모델을 적용하여 투자 의사 결정 모델을 개발한다. P2P 대출 시장에서 대출 상태는 불균형한 데이터이기 때문에 IEFSVM을 적용하면 완납된 대출을 예측할수 있다. 또한, 수익성을 높이기 위해 다중 회귀 분석 모델을 사용하여 높은 투자 수익을 가지고 파산하지 않을 대출을 찾는다. 흥미롭게도 IEFSVM은 분류 성능 측면에서도 기존의 클래스 불균형 분류기를 개선하고, 수익성 성과와 관련하여서도 투자 의사 결정 모델을 개선하는 데에 성공한다. 결론적으로, 이 논문의 기여도는 새로운 비용 민감 분류기의 개발과 수익성 있는 투자 결정을 위한 분류기의 응용을 포함한다.

주요어: 퍼지 서포트 벡터 머신, 엔트로피, 정보 이론, 최근접 이웃, 클래스 불균형 분류, P2P 대출 시장, 투자 결정, 대출 상태 예측

학번: 2013-21083

감사의 글

어릴 적부터 인복이 좋다는 이야기를 참 많이 들었던 것 같습니다. 제가 박사 논문을 이렇게 마무리 지을 수 있었던 건 절대 제 노력만으로 이루어진 것이 아니라 주위 분들의 도움과 헌신 덕분이라고 말할 수 있습니다. 항상 감사하며 보은하고 베풀며 살아 가도록 하겠습니다.

우선, 제 지도교수님이신 장우진 교수님께 깊은 감사를 드립니다. 제 인생의 선택 중 제일 잘한 것은 교수님께 박사과정 지도를 받은 것입니다. 연구에 대해 아무것도 모르던 저를 실수가 있어도 넓은 마음으로 받아주시고, 연구자로서의 길을 가르쳐주신 덕분에 제대로 된 연구를 할 수 있었습니다. 또한, 논문 지도 외에도 인생에 필요한 지혜를 알려주시고 항상 저를 지지해주신 모든 것에 감사드립니다. 항상 감사한 마음 잊지 않고, 그 마음에 보답하는 사람이 되도록 하겠습니다. 그리고 바쁘신 가운데에도 제 학위논문의 심사위원장을 맡아주신 이재욱 교수님과 심사하는 과정에서 아낌없는 지도를 해주신 이덕주 교수님, 박건수 교수님, 박철진 교수님 네 분께 모두 감사드립니다. 날카로운 지적과 따뜻한 조언들이 있었기에 학위논문을 무사히 마무리 할 수 있었습니다. 정말 감사드립니다.

연구실에 오랜 기간 동안 터줏대감으로 있으면서 정말 많은 사람과 소중한 인연을 맺을 수 있었습니다. 소중한 자산이라 생각하고 정말 감사합니다. 많은 일들이 있었고, 돌이켜보면 인생에 있어 가장 행복한 순간들의 연속이었습니다. 과거로 돌아갈 수 있다면 당당히 연구실에 입학했을 때로 돌아가고 싶다고 말할 수 있습니다.

먼저, 연구실 생활에 대해 많은 조언을 해주신 경원이 형. 연구실 생활 충분히 즐기고 준비되어 사회로 나가는 것 같습니다. 앞으로도 좋은 이정표가 되어 저희를 잘 이끌어 가주시리라 믿습니다. 항상 인자한 웃음으로 반겨주시던 호진이형. 졸업하시고 나서 회사에서 힘들어 보이기도 하시고 많이 뵙지 못했지만, 술한잔 하자고 호진이형한테 칭얼거리던 때가 그리워요. 그리고, 송 교수님 그저 감사합니다. 앞으로 보은하며 살아가도록 하겠습니다. 연구도 계속 이어나갔으면 좋겠습니다. 가장 재미있는 술자리는 션이 있는 자리. 많이 참석하지 못해 죄송해요. 그래도 계속 불러주실거죠? 언제나 유쾌한 승민이 형. 같이 당구치고 공모전 했을 때가 그립네요. 결혼 축하드려요. 모든 게 앞서나간 봉균이 형, 타지에서 힘드시겠지만 앞으로도 많이 만날 수 있었으면 좋겠어요. 연구가 어울릴 것 같아서 박사로 돌아오라고 계속 얘기했던 강원이 형, 한화맨이 되어 가장 행복한 삶을 살고 계시네요. 제일 오랫동안 수업 같이 들으면서 항상 고마웠던 민혁이. 너의 우직한 성향 계속 밀고 나가길 바래.

텝스 공부를 마치고 논문 준비할 때에는 남은 형들한테 참 죄송한 점이 많았네요. 투정 부리고 예민하게 구는 것 다 받아주시고, 정작 저는 큰 도움이 된 것 같지 않아 미안한 마음 뿐이에요. 처음엔 어떻게 박사까지 하시려나 걱정됐는데, 정말 많이 의지가 된 우리 연구실의 아이콘 창주형. 연구하라고 계속 구박만 한 것 같지만, 박학다식한 선도형. 졸업만 하면 특유의 일 능력으로 날아다니지 않을까요. 어렵다고만 생각했지만, 은근히 비슷한 면이 많은 지환이 형. 닮고 싶었던 점이 참 많았던 것 같아요. 텝스 공부할 때 멘탈 관리 시켜준 승모형, 많은 힘이 되었습니다. 연구실의 얼굴 성윤이형. 박사과정 잘 준비하시길 바래요. 모두들 참 고맙고, 얼른 졸업하시고 사회에 와서 만나요.

나보다 어린 친구들에게도 많은 도움이 되고 싶었지만, 졸업 준비에 치이느라 그러지 못한 것 같다. 똑똑하기만 했던 동규, 갈굼도 많이 받았지만 훌륭한

선배가 된 것 같아 대견스럽고, 연구실을 잘 이끌어 갈 것이라 믿는다. 강한 송정윤. 같이 운동할 사람이 없어서 심심하겠다. 수업 적당히듣고 얼른 졸업하길 바래. 상상력 많은 득화야. 카추사 나온 만큼 어딜 가서든 눈치 보지말고 너의 의견을 피력하렴. 항상 선배들한테 싹싹하던 도현이. 뜻하는 바 이루길 바라고, 널 위해 언제든 술 한잔 기울일 수 있단다. 병훈이는 만난 시간은 짧지만, 멋진 꿈을 펼칠 수 있길 바래. 현주한텐 항상 미안하고, 나를 포함해서 우리 연구실 사람들을 다 포용할 수 있는 관대함을 가지고 있으리라 믿고 있다. 준열이와 우혁이는 만난 시간은 짧지만 잘 할것이라 믿고, 남은 선배들에게 많은 것들을 배우며 도와주길 바란다.

마지막으로, 이 모든 것을 가능하게 해주신 부모님께 영광을 바칩니다. 어려운 결정이셨을텐데, 큰 아들 끝까지 믿고 제 결정 존중해주셔서 감사합니다. 부모님 덕분에 아무 걱정 없이 마음껏 배우고, 놀고, 행복한 추억을 간직하며 자랄 수 있었습니다. 아무리 힘든 일이 있어도 행복한 가족이 있어 버틸 수 있었습니다. 고등학교 때부터 나와 살아서 걱정도 많이 되게 하고, 효도도 제대로 못 해드린 것 같아 마음 한 켠이 무겁습니다. 저를 걱정하시는 것도 이제 졸업입니다. 빽만하면 경기를 일으키던 7살 아이는 이제 꿈을 이룬 29살 청년이 되었습니다. 이제는 제가 더 챙겨드리고 행복하게 해드리겠습니다. 동생 호진이에게는 내가 형의 역할을 많이 못해준 것 같아 미안하구나. 못난 형을 용서하고, 바라는 꿈 이뤄지기를 응원한다. 모두 감사합니다.

2019년 7월

조 풍 진