博 士 学 位 論 文 要 約

論 文 題 目: Improving Classification Accuracy for Machine Learning

（機械学習における分類精度の向上）

氏　　名： 鄭　弯弯

要　　約：

This thesis is organized under five chapters. Chapter 1 gives a brief explanation of what machine learning is and why it matters. Chapter 2 makes a proposal to improve the performance of feature selection methods with low-sample-size data. Chapter 3 studies the effects of class imbalance and training data size on classifier learning empirically. Chapter 4 proposes a fast noise detector referring to the problems of noise detection algorithms, which are over-cleansing, large computational complexity and long response time. Chapter 5 draws a summary and the closing.

1. Introduction: what machine learning is and why it matters

Machine learning refers to a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data or other outcomes of interest, or to perform other kinds of decision making under uncertainty. Especially in the era of big data, the information we are drowning in has gained prominence so that

it becomes impractical for scientists to handle it humanly. Machine learning enables analysis of massive quantities of data, which can be numbers, words, images, voice and clicks, what have you. That way, it has become an important aspect of modern business and research, and powers many of the services we use today, such as recommendation systems like those on Amazon, Netflix and Watson; search engines like Google, Baidu and Bing; social–media like Twitter, TikTok and Facebook; voice assistants like Siri, Cortana and Nina. This list goes on.

Machine learning is not a new science (since 1949 when D. Hebb created a model of brain cell interaction) but has gained fresh momentum in the current era, and still faces numerous problems. Among them, high–dimension, low–sample–size data, class imbalance and noise/outlier are highly emphasized. This study tried to improve the classification accuracy of machine learning from feature selection, classifier selection and noise detection regarding to the three challenges, respectively (see Figure 1).

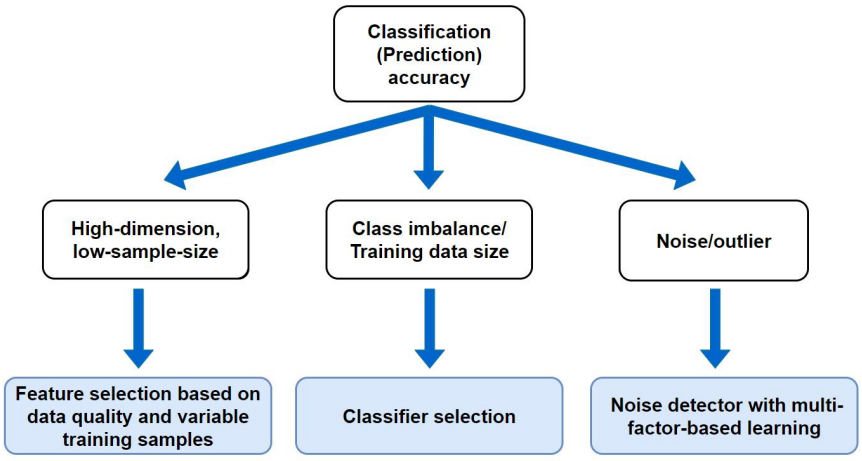


Figure 1: The configuration of this study.

2. A proposal to improve the performance of feature selection methods with low-sample-size data

2.1 Introduction

According to feature selection methods, sufficient samples are usually required to select a reliable feature subset. In a dataset with a considerable number of samples, the effects of outliers will be limited, and the training data will represent the population at large. However, with low-sample-size data, the values of few outliers can significantly convert the set of selected features into a new set of potential noisy features that may not fully reflect or capture class-specific differences. Furthermore, though conventional feature selection adopts random sampling to improve the performance, low-sample-size datasets are typically too small to be processed using this method. Datasets characterized by a small number of samples are common in various areas (e.g. studies of rare diseases or extraordinary athletes).

This study aimed to propose a novel approach (Feature Selection Based on Data Quality and Variable Training Samples, QVT) to handle the feature selection with low-samplesize data.

2.2 Methodology

The QVT refers to a two-phase hybrid approach. Because the performance of feature selection methods is affected by the quality of data and the number of samples, to improve the performance of feature selection with limited data, the first phase is to

define the most typical samples of each class. In the second phase, feature selection starts from using the most typical samples, which is repeated with a steady increase in sample size until all of samples are used. In this process, the list of selected features would be kept updated. Figure 2 is used to explain the image of QVT.

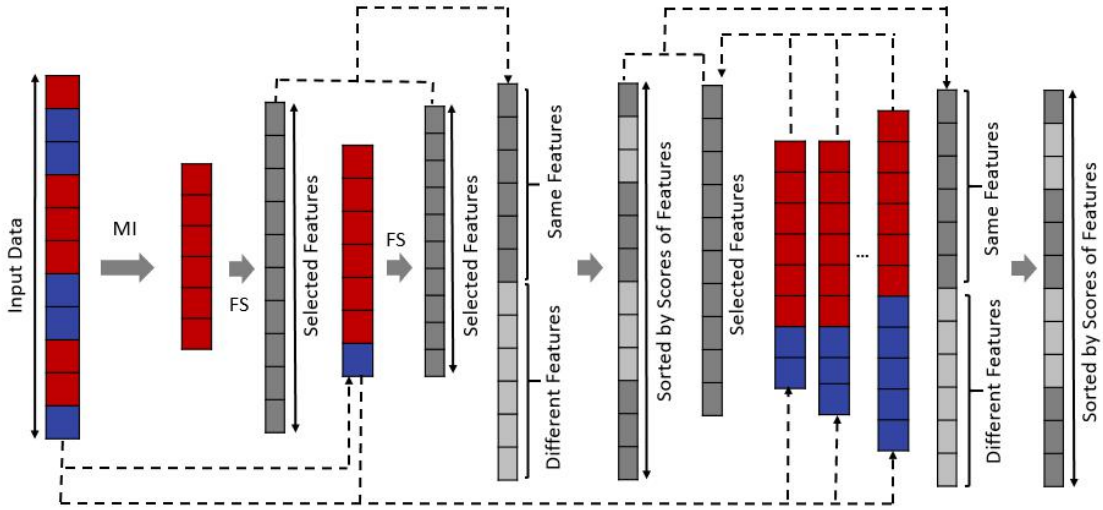


Figure 2: The image of QVT.

2.3 Experiments study

2.3.1 Baselines, classifiers and performance metrics

In this study, IG, Boruta and Lasso were taken to implement QVT for they are different methodologies. In most of previous research, because the number of selected features was primarily determined by researchers subjectively, how to determine the optimal number of selected features is still a problem in feature selection study. IG, Boruta and Lasso address this problem by automatically eliminating irrelevant features not associated with the classification task, making the problem mentioned above easier. Besides, synthetic data generation method ANS was chosen as another baseline. ANS

was originally proposed as an oversampling technique for class imbalance problem. Because of its efficiency and parameter-free characteristic, ANS has proven successful in variety of applications from several different domains, such as face recognition, software engineering and medical diagnosis.

In this study, Support Vector Machine (SVM), Naïve Bayes (NB) and Logistic Regression (LR) were adopted as classifiers. the performance of classifiers was evaluated by two metrics: macro-F (macro-averaged F measure) and AUC (Area Under the ROC curve).

2.3.2 Experiments

Five steps were performed.

(i) 2/3 samples were extracted from each class randomly as training data, and the remaining samples were used as test data. Furthermore, using ANS to increase the training data size of each class to five times.

(ii) Boruta, ANS(B), QVT(B), IG, ANS(I), QVT(I), Lasso, ANS(L) and QVT(L) were adopted to perform feature selection for training data, respectively. Subsequently, the selected features were sorted according to the importance score. The most important feature was ranked on the top.

(iii) Increasing one by one from two features to train classifiers (SVM, NB and LR). Then, test data were predicted, and macro-F and AUC were computed each time.

(iv) Six measures were adopted to evaluate the effectiveness of feature selection methods, which are explained as follows.

• The lowest value of performance metric (Min.) and the greatest value of performance metric (Max.).

• Because after implementing ANS and QVT, the number of selected features might be different, the other two measures are the value using the same number of selected features (Sam.) and the value using all selected features (All). About the computation of Sam., for instance, X is considered a feature selection method. For X, ANS(X) and QVT(X), the numbers of selected features are *a*, *b* and *c* (a is the smallest), respectively. Sam. was obtained using the top *a* features selected by X, ANS(X) and QVT(X), respectively.

• The macro–F and AUC of X, ANS(X) and QVT(X) were compared after increasing the number of features. Lastly, the win times of X, ANS(X) and QVT(X) were counted (#Win). Here, the case of tie win would not be counted. Furthermore, the average number of used top features (mRank) when X, ANS(X) or QVT(X) won was computed.

For instance, hypothesize X, ANS(X) and QVT(X) selected 4, 8 and 6 features, respectively. Consider the notion V(f) represents the value of performance metric (V) and the number of features used (f). For X, the results are 0.90(2), 0.93(3), 0.91(4); For ANS(X), they are 0.93(2), 0.93(3), 0.94(4), 0.95(5), 0.89(6), 0.89(7), 0.89(8); For QVT(X), they are 0.89(2), 0.93(3), 0.95(4), 0.95(5), 0.91(6). Because 0.93(2) > 0.90(2) > 0.89(2),

win tie happens with the case of top 3 features, 0.95(4) > 0.94(4) > 0.91(4), win tie happens with the case of top 5 features, 0.91(6) > 0.89(6), The #Wins of X, ANS(X) and QVT(X) will be 0, 1 and 2, the #mRank NA, 2 (2/1=2) and 5 (10/2=5), respectively.

Because the lower the mRank, the better the feature selection method would be, X, ANS(X) or QVT(X), which gets lower mRank value will win. Min., Max., Sam., All and #Win are contrary to mRank.

(v) After step1 ~ step4 were performed ten times, the average of Min., Max., Sam., All, #Win and mRank were taken and considered as the final evaluation measure of X, ANS(X) and QVT(X).

2.3.3 Experimental results

Figures 3 shows the percentages of win, loss and tie of QVT(B) compared with Boruta within six evaluation measures for all datasets. Because we have 20 datasets, 3 classifiers and 2 performance metrics, the percentages shown in figure were obtained using the results of 120 cases. According to Figures 3, the win probabilities of QVT(B) were 68%, 64%, 51% and 53% when the training samples were 13, 20, 27 and 33, respectively. For QVT(I), the win probabilities were 72%, 68%, 66% and 62%, respectively; for QVT(L), they are 78%, 76%, 62% and 78%, respectively. This is same as the average results showed, i.e. the superiority of QVT was weakened with the increase in training data size. However, the least percentage of win 51% was still over 50%. When compared with ANS, ANS showed great performance, however, the

weakness was also easy to see, i.e. the smaller the training data size, the weaker the learning ability of ANS. According to the results, ANS seems not a good choice for data with less than 13 samples.

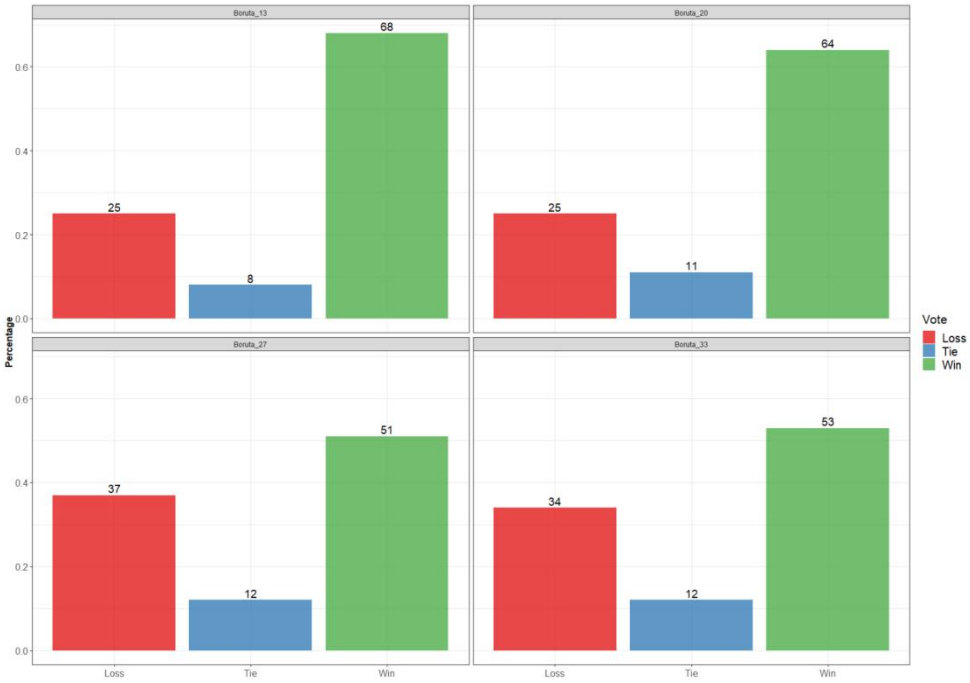


Figure 3. The percentages of win, loss and tie of QVT(B) within six evaluation measures for all datasets (compared with explicit feature selection methods).

As conclusions, QVT can improve the performance of different feature selection methods. Additionally, QVT showed to be more robust against the low-sample-size than ANS. Furthermore, because the performance of all classifiers were improved after implementing QVT compared with that using explicit feature selection method and ANS, in conclusion, QVT has the similar effect on different classifiers.

2.4 Conclusions

In this study, QVT was proposed to improve the performance of feature selection methods with low-sample-size data. According to the experiment results: (1) the

feature selection methods fit a classification problem with less than 33 training samples; (2) a smaller number of training samples led to a more significant difference between QVT and the baselines, and QVT was verified as the better one; (3) QVT fits different feature selection methods, and it can significantly improve the predictive performance of different classifiers; (4) QVT shows to be more robust to handle feature selection of low–sample–size data than the synthetic data generation method ANS.

3. The effects of class imbalance and training data size on classifier learning: an empirical study

3.1 Introduction

Class imbalance is one of the most serious influential factors for the predictive performance of classifiers. The imbalanced data are characterized as having more instances of certain classes than others. In this case, classifiers tend to make a biased learning model that has a poorer predictive accuracy over the minority classes compared to the majority classes. This is because most standard classifier learning algorithms, such as decision tree, backpropagation neural network and support vector machines, are designed based on assumptions that the class distribution is relatively balanced and the misclassification costs are equal, classification rules that predict the minority classes tend to be rare, undiscovered or ignored. Consequently, test samples belonging to the minority classes are misclassified more often than those belong to the majority classes. It is said that classification of data with imbalanced class distribution

has encountered a significant drawback of the performance attainable by most standard classifier learning algorithms.

In classification studies, the more powerful machine learning algorithms should be able to learn complex nonlinear relationships between input and output features. By definition, they are robust to noise, show high variance, and meaning predictions vary based on the specific data used to train them. This added flexibility and power comes at the cost of requiring more training data, often a lot more data. Therefore, in studies of machine learning, collecting more data should be the most common and easiest way to improve performance of classifiers to a desired level. However, in numerous real–world applications, the number of samples in a dataset can be relatively limited, such as in studies of rare diseases, extraordinary athletes or medical images, which often have limited sample size restricted by the availability of the respondents as well as the robotics application due to the cost of the data collection process.

This study compared a total of twelve classifiers, attempted to clarify the effects of class imbalance and training data size on classifier learning.

3.2 Datasets, classifiers, and performance metric

This study used nine benchmark datasets extracted from different domains to draw a relatively universal conclusion. Additionally, twelve representative classifiers arising from seven categories were selected and the hyperparameter tuning was done for every data to maximize the performance of each classifier. Furthermore, to make the result

more reliable, macro–F was chosen to measure the predictive performance of classifiers.

3.3 Experiments

In this study, the degrees of class imbalance were set as 1.00, 0.85, 0.70, 0.55, 0.40, 0.25 and 0.10. Although the data characteristics enable different classifiers to perform very well or very baldly, according to the results, no outlying dataset exists among the nine datasets. Therefore, the average macro–F across nine datasets was taken and is shown in Figure 4. Ensemble classifiers AdaBoost, XGBoost, parRF and RF were superior. The avNNet, C5.0 and svmPoly were placed on the second–class level of performance. In addition, LLM, svmRadial, CART, NB, and LR were less susceptible to data imbalance compared with the other applied classifiers. Especially, the performance of svmRadial kept being improved with the mitigation of imbalance rate. All of classifiers obtained the lowest macro–F when the skew was 0.10. However, after the skew is larger than 0.40, the effect of imbalance seemed to be eliminated through hyperparameter tuning.

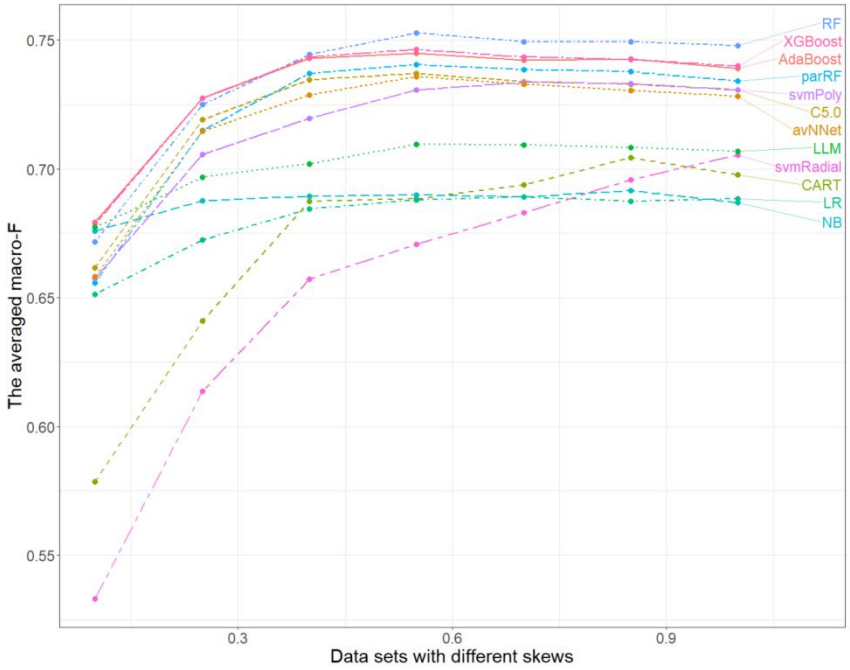


Figure 4. Averaged macro–Fs on test data of nine datasets.

Furthermore, Tukey—Kramer test was performed to find whether there were significant differences between classifiers in the averaged macro–F and the coefficient of variation (CV). The results indicated that the predictive performance of CART and svmRadial was significantly inferior than that of the other classifiers at 0.05 level. Because in the case of averaged macro–F, significant differences were detected between CART and AdaBoost (p=0.02), CART and RF (p=0.02), CART and XGBoost (p=0.03), svmRadial and AdaBoost (p=4E–04), svmRadial and avNNet (p=0.01), svmRadial and C5.0 (p=7.54E–03), svmRadial and parRF (p=2.19E–03), svmRadial and RF (p=2.7E–04), svmRadial and svmPoly (p=0.01), svmRadial and XGBoost (p=4.4E–04). Furthermore, according to the results of Tukey–Kramer test on CV, no significant difference exists between CART and the other classifiers, except LR. On the other hand, the CV of LR was around 0.20, which was significantly larger than those of AdaBoost (p=1.8E–06), avNNet (p=0.01), C5.0 (p=0.03), NB

(p=0), CART (p=5.6E−04), LLM (p=1.7E−04), parRF (p=3.6E−05), RF(p=4E−06), svmPoly (p=2.94E−03), svmRadial (p=0.01) and XGBoost (p=0.03) at 0.05 level. As we can see, LR exhibits a problematic behavior in terms of predictive accuracy and is more highly dependent on the data characteristics, although the predictive performance of all classifiers is affected to some extent.

3.4 Conclusions

In this comparative study we clarified the effects of class imbalance and training data size on the predictive performance of classifiers. Twelve frequently employed classifiers were selected, and thorough hyperparameter tuning was performed. The predictive accuracy (macro−averaged F measure) and the rank of classifiers were studied and the results indicate that (1) NB, LR and LLM are less susceptible to class imbalance while they have relatively poor predictive performance; (2) Ensemble classifiers AdaBoost, XGBoost, RF and parRF have a quite poorer stability in terms of data imbalance while they achieved superior predictive accuracies; (3) avNNet, C5.0, svmRadial and svmPoly are placed on the second−class level of performance; (4) CART and NB are the last choices among the twelve classifiers; (5) For all of the classifiers employed in this study, their accuracies decreased as soon as the class imbalance reached a certain point 0.10. Note that although using datasets with balanced class distribution would be an ideal condition to maximize the performance of classifiers. In the case that the skew is larger than 0.10, a comprehensive hyperparameter tuning may be able to eliminate the effect of imbalance.

4. A fast class noise detector with multi-factor-based learning

4.1 Introduction

Noise detection is a preprocessing task that can be employed in any given dataset to identify potentially noisy instances. It is suggested that some objects can be impacted by feature or class noise, which offer misleading information and then hinder the learning process of classifiers. Moreover, class noise is potentially more harmful than feature noise, and the reasons are presented below. First, there are numerous features, whereas there is only one label. Second, each feature for learning has different importance, whereas labels always significantly impact learning. Third, the consequences of class noise detection will impact many feature noise detection techniques (e.g. feature selection) directly. Thus, detecting class noise prior to the analysis of polluted data appears to be necessary. In noise detection field, over-cleansing and the high time complexity are primarily difficult to solve. This study proposed a fast class noise detector with multi-factor-based learning, which is capable of alleviating the possibility of over-cleansing and performing favorably to three existing ensemble based techniques and the original data in terms of classification accuracy. Moreover, the fast execution makes it possible to deal with large-scale datasets.

4.2 The FMF algorithm

FMF is a three-stage process. Because no method can be efficient for all of data, the first stage determines the proper similarity index for the applied data. The second stage exploits the determined similarity index to make a multi-factor-based learning to yield a noise score for each instance. In the third stage, a threshold is given to remove the noisy instances. The workflow of FMF is illustrated in Figure 5.

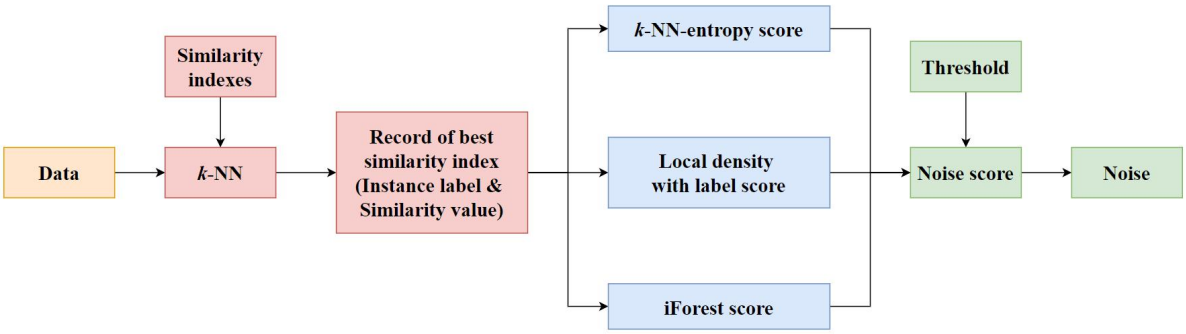
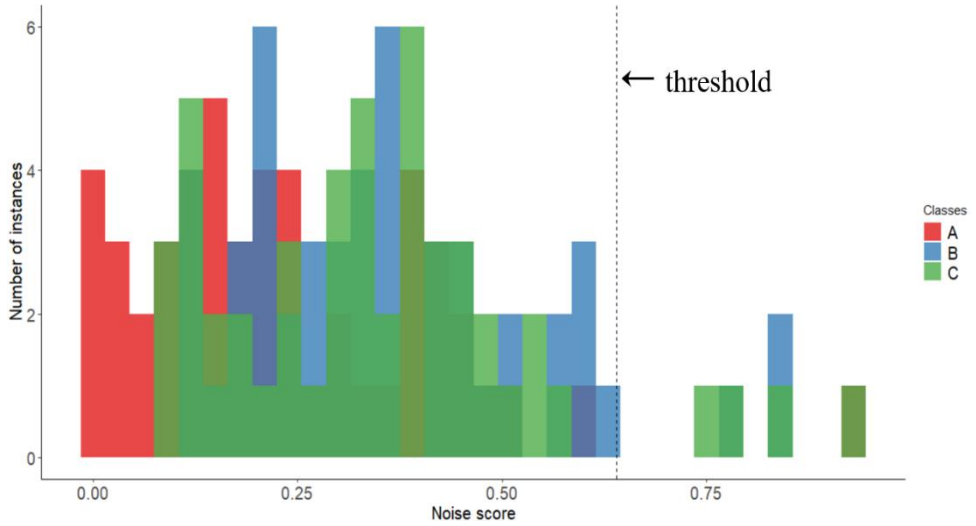


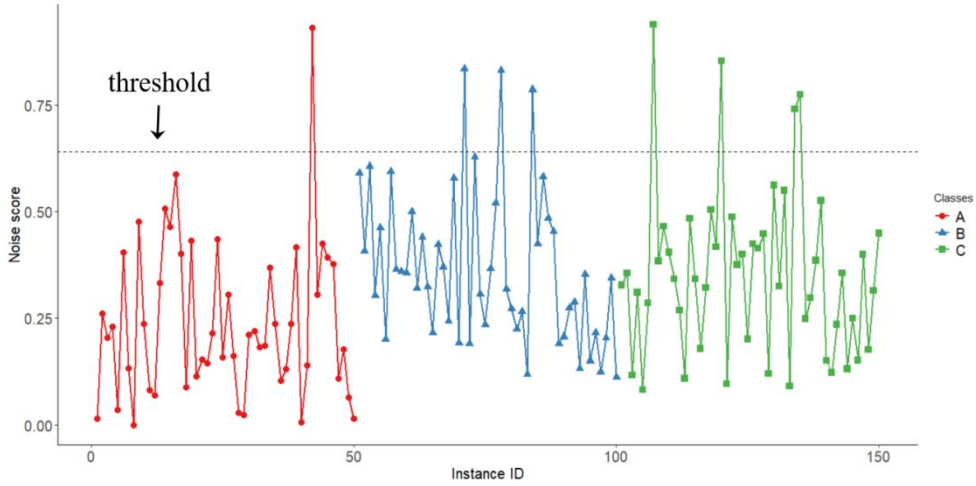
Figure 5. The workflow of FMF.

4.3 Noise detection with FMF

Figure 6 presents the distribution of noise scores and the noise score of each instance. For iris, the minimum and maximum of noise scores were 0 and 0.94, respectively. The reduction rate was about 5.33% (≈ 100 × 8/150) referring to threshold 0.63. Furthermore, the detected noisy instances distributed at the right with higher values and had different distribution (in location and distributional form) from the other instances.



(A) Distribution of noise scores



(B) Noise score of each instance

Figure 6. The distribution of noise scores and the noise score of each instance (iris, the best similarity index: Euclidean).

Figure 7 presents the result of Principal component analysis (PCA) of iris to visualize the relationship between the position and noise score. For each class, a color gradient from blue to red specifics the data points positioned in the center zone, middle zone and border zone. Three points must be remarked:

(i) The instances that completely enter another class (e.g., data points with ID 107 and 120) and the outliers (data point with ID 42) got the maximum scores, showing the maximum possibility to be the noise.

(ii) The data points around the decision boundary (e.g., data points with ID 71, 84, 135) got median scores, showing median possibility to be the noise.

(iii) The central data points of each class get the minimum scores, showing the maximum possibility to be the normal instances (or representative instances).

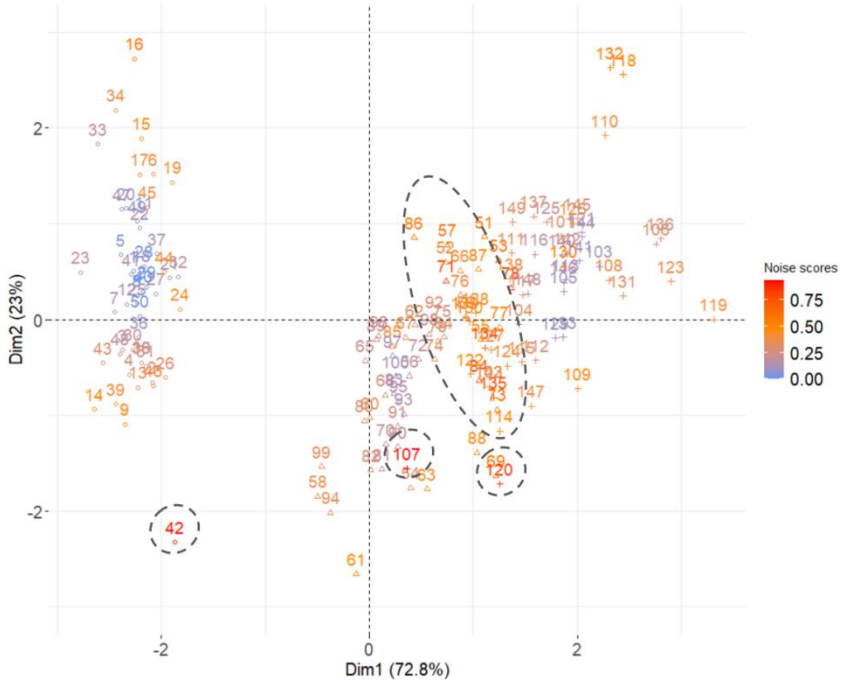


Figure 7. Principal component analysis (PCA) of iris data.

4.4 Comparison between FMF and the baselines

FMF was compared with the case of using the original data, All *k*NN, dynamicCF and INFFC. To assess the accuracy achieved by the mentioned algorithms, besides 5–NN, SVM and C5.0 were also applied. Furthermore, because noise detection is a multi–objective optimization problem, the classification accuracy, the reduction rate and the processing time are shown to verify the hypothesis of the enhanced performance of FMF statistically.

The results are summarized as follows:

• About the classification accuracy, removing instances identified by FMF for training achieved the highest overall classification accuracy compared with the classifiers trained on the entire training data as well as with noise removed by the other methods, revealing that FMF detected the noisy instances successfully.

• About the reduction rate, FMF reached the lowest value. Compared with All *k*–NN which achieved the maximum reduction rate, the value of FMF was about its 1/3. One defect of FMF is the reduction rate, whereas the low reduction rate also demonstrates the low possibility of over–cleansing.

• About the processing time, FMF efficiently sped up the computation. Compared with the 2nd fast dynamicCF and the slowest INFFC, its speed was about 20 times and 333 times faster, respectively. Accordingly, FMF is considered capable of processing the large–scale datasets as well as being applicable to the problems that required to be taken care of timely (e.g. the fraudulent use of credit cards and an unusual computer network traffic).

4.5 Conclusions

This study proposed a fast noise detector with fundamentally different structure that makes a multi–factor–based learning rather than the normal simple learning (e.g., density–based learning) or ensemble based learning with classifiers. Removing the

noisy instances which completely enter other classes first and then smoothing the decision boundary allows FMF to achieve a promising result.

The empirical evaluation showed that FMF reached better results and retention than the state–of–the–art ensemble–based methods All *k*–NN, dynamicCF and INFFC in terms of macro–F and processing time, especially for large datasets. Moreover, it obtained the competitive results compared with the use of original data.

5. Summary

This study tried to improve the classification accuracy from feature selection, classifier selection and noise detection regarding to the three main challenges in machine learning, i.e. high–dimension, low–sample–size data, class imbalance and noise/outlier. About the problem of handling high–dimension, low–sample–size data, a method was proposed to improve the performance of feature selection methods using the original data, which is named Feature Selection Based on Data Quality and Variable Training Samples (QVT). According to the experiment results, a smaller number of training samples led to a more significant difference between QVT and the baselines, and QVT was verified as the better one. The classification accuracy was improved up to 13%. About the problem of class imbalance, the effects of class imbalance and training data size on the predictive performance of classifiers was discussed through an empirical study. About the problem of noise/outlier, this study proposed a fast class noise detector with fundamentally different structure that makes a multi–factor–based learning (FMF) rather than the normal simple

learning (e.g., density–based learning) or ensemble–based learning with classifiers. According to the experiment results, removing instances identified by FMF for training achieved the highest overall classification accuracy compared with the classifiers trained on the entire training data as well as with noise removed by the other methods. The classification accuracy was improved up to 18%; furthermore, compared with the second fast dynamicCF and the slowest INFFC, its speed was about 20 times and 333 times faster, respectively.

Many machine learning algorithms have been around for a long time and researchers never stop creating more powerful algorithms. Thus, the ability to automatically apply more complex, more effective and more robust mathematical calculations to data is kept being improved over and over. That way, the useful information is extracted more timely and works better for human beings. Although in current status, it is still common to get higher accuracy at the cost of higher computational complexity, and the universal 100% accuracy has not been achieved, classification algorithms which are accurate enough and fast enough must be created in the future and we are on the way.