Computer and Information Science; Vol. 8, No. 1; 2015
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

An Empirical Analysis of Imbalanced Data Classification

Shu Zhang[1], Samira Sadaoui[1] & Malek Mouhoub[1]

[1] Department of Computer Science, University of Regina, SK, Canada

Correspondence: Samira Sadaoui, Department of Computer Science, University of Regina, SK, Canada. E-mail: sadaouis@uregina.ca

Received: December 15, 2014 Accepted: January 3, 2015 Online Published: January 29, 2015

doi:10.5539/cis.v8n1p151 URL: http://dx.doi.org/10.5539/cis.v8n1p151

Abstract

SVM has been given top consideration for addressing the challenging problem of data imbalance learning. Here, we conduct an empirical classification analysis of new UCI datasets that have different imbalance ratios, sizes and complexities. The experimentation consists of comparing the classification results of SVM with two other popular classifiers, Naive Bayes and decision tree C4.5, to explore their pros and cons. To make the comparative experiments more comprehensive and have a better idea about the learning performance of each classifier, we employ in total four performance metrics: Sensitive, Specificity, G-means and time-based efficiency. For each benchmark dataset, we perform an empirical search of the learning model through numerous training of the three classifiers under different parameter settings and performance measurements. This paper exposes the most significant results i.e. the highest performance achieved by each classifier for each dataset. In summary, SVM outperforms the other two classifiers in terms of Sensitive (or Specificity) for all the datasets, and is more accurate in terms of G-means when classifying large datasets.

Keywords: imbalanced data classification, SVM, Naive Bayes, decision tree, empirical search, performance metrics

1. Introduction

Data classification is a significant research topic in the areas of data mining and machine learning. There are two major tasks in data classification: the first one is to learn from the training sample with given labels, and the second one is to employ the learned knowledge to assign a label to an unknown sample (Cherkassky Mulier, 2007). A well-known (binary) classifier is the Support Vector Machine (SVM), which was initially introduced by Vapnik (Vapnik, 1998). SVM, a relatively new machine learning method, became very successful due to its strong theory and excellent performance in data classification (Haibo Garcia, 2009; Boolchandani Sahula, 2011). Additionally, SVM has shown remarkable success in various domains, such as pattern recognition and text classification. For the reasons mentioned above, SVM has been given top priority for addressing the challenging problem of imbalanced data, i.e., when the majority class is much larger than the minority class, or vice versa (Cao et al., 2013). The imbalance learning problem is observed in many areas, like fraud detection and medical diagnosis. For instance, in online auctions, fraud activities are fewer than normal activities. In fact, online auctions have skewed user datasets (normal vs. fraudsters). It is hard to classify the abnormal users, but it is significantly important to detect them (Mamum Sadaoui, 2013). Fraudulent behaviours, such as shill bidding and bidder collusion, usually result in a negative impact on honest users such as money loss and waisted effort (Mamum Sadaoui, 2013). Learning from training data that are imbalanced is diffcult since the standard machine learning systems often misclassify minority instances as majority ones (Koknar-Tezel Latecki, 2009). This means that the prediction of classifying a new data into the minority class is very low (Haibo Garcia, 2009). In many real-life situations, the cost of incorrectly classifying the minority event is much higher than misclassifying the normal event, for example misclassifying a fraudulent bidder as an honest one. As a result, many researchers developed various techniques to correctly classify imbalanced datasets, such as data sampling methods that have been proven to improve the classifier performance (Koknar-Tezel Latecki, 2009; Haibo Garcia, 2009). On the other hand, the imbalance problem is not the only factor that impacts the classification accuracy, but the data complexity plays a more significant role (Haibo Garcia, 2009).

In this study, we select five relatively new datasets from the UCI repository (the machine learning data centre)(UCI, 2013) as there are no previous works that conducted the classification experiments on these data to the best of our knowledge. In order to make the empirical analysis more interesting, the selected datasets have different imbalance ratios (from slightly to extremely imbalanced), different sizes (in terms of the number of instances) and different complexities (in terms of the number of attributes). Regarding SVM, the kernel function selection and parameter setting have a great impact on the classification results. In our point of view, experimentation is a very good approach to determine which combination of a kernel and a parameter setting lead to the best performance. An empirical search of the learned model, consisting of the best kernel function and parameter setting (with K-fold cross validation and penalty C) is performed through many training of SVM under several performance measurements. Geometric-means has been utilized as the performance metric for the classification problem of class imbalance (Imam, et al., 2006). We also pay attention to the Sensitive metric since it provides the accuracy for the minority class samples, which we are most concerned with when classifying imbalanced data. According to (Akbani, et a., 2004), SVM performs not too badly for moderately imbalanced classes when compared to other machine learning techniques, but fails on highly imbalanced data. Hence, before applying SVM, we first pre-process each dataset in order to improve the classification results, such as transforming a multi-class problem into a binary one, and sampling those datasets that are highly imbalanced. Furthermore, we compare the classification results of SVM with two other famous classifiers, Naive Bayes and Decision Tree C4.5, to explore their pros and cons. To make the comparative experiments more comprehensive and have a better idea about the learning performance of each classifier, we employ in total four metrics: Sensitive, Specificity, G-means and time-based efficiency.

The following sections are organized as follows. Section 2 introduces the principles of SVM and the kernel functions as well as several performance metrics. Section 3 discusses existing classification solutions for imbalanced data. Section 4 presents the empirical analysis approach we adopted for classifying imbalanced data. Section 5 conducts in detail various experiments on five real-world datasets by using three supervised learning techniques: SVM, Naive Bayes and decision tree (C4.5), and discusses their classification results followed by a comparative analysis. Section 5 only exposes the most significant results i.e. the highest performance achieved by each classifier for each dataset under each metric. Finally, Section 6 concludes this paper and presents one future research direction.

2. Background

In this section, we introduce the concepts of the standard SVM as well as the kernel functions to overcome the issue of linearly inseparable data. In addition, we present various measurements for evaluating the performance of the classification techniques.

2.1 Support Vector Machine

We explain the basics of SVM via the example depicted in Figure 1. Firstly, we introduce the concept of hyperplane (denoted with a solid line). Particularly, there are two linear hyperplanes in Figure 1: one is inclined and the other one is horizontal. Both separate the samples into two classes: triangle instances and square instances. The concept of margin indicates the distance or gap between the hyperplane and a data instance (Cortes Vapnik, 2014). The ultimate goal of the SVM model is to search for the optimal hyperplane that maximizes the margin between the nearest instances and the hyperplane, and at the same time minimizes the classification errors (Cortes Vapnik, 2014). These instances are called boundary instances, or support vectors, and are illustrated with circles. In other words, maximizing the distance between data of both classes is the core idea of SVM. So, in our example, SVM prefers the horizontal hyperplane since it possesses a larger margin between the two data categories.

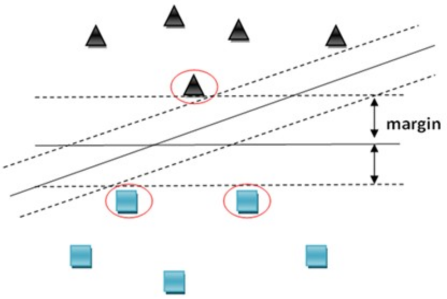

Figure 1. Standard SVM

Thanks to the kernel functions, SVM can address the complex issue of linearly inseparability of data (Hsu, et al., 2010). In most real-life applications, data are not always linearly separable, consequently we need to find a non-linear classifier. It has been shown that SVM can always find a hyperplane that linearly separates the samples in a certain high dimensional space for the linearly non-separable input data in the low dimensional space (Cortes Vapnik, 2014). Feature mapping is utilized to transfer this problem from low dimensional space to a higher dimensional one. For example from two dimensional linearly un-separable into three dimensional linearly separable (Cortes Vapnik, 2014). Subsequently, from the three dimensional feature space, a kernel function can be performed to obtain the non-linear classifier for the original two dimensional space.

Table 1 exposes several types of classical kernels; $K(X_i, X_j)$ is the kernel that measures the closeness of two support vectors. In our experiments, γ is one of the parameters that we need to adjust to reach good results of data classification. γ shows how far is the impact of a training sample, with a low value meaning 'far' , and a high value meaning 'close' (Muller, et al., 2002). A value of γ determines a boundary, for instance when γ is large, the RBF function is very narrow.

Table 1. Kernel functions

Type	**Formula**
Linear Kernel (Hsu, et al., 2010)	$K(X_i, X_j) = X_i^T X_j$
Polynomial Kernel (Hsu, et al., 2010)	$K(X_i, X_j) = (\gamma X_i^T X_j + r)^d, \gamma > 0$
Radial Basis Function Kernel (RBF) (Hsu, et al., 2010)	$K(X_i, X_j) = \exp(-\gamma \lVert X_i, X_j \rVert^2), \gamma > 0$
Sigmoid Kernel (Hsu, et al., 2010)	$K(X_i, X_j) = tanh(\gamma(X_i^T(X_j + r)$
Power Kernel (Boolchandani & Sahula, 2011)	$K(X_i, X_j) = -(\lVert X_i - X_j \rVert^\beta), 0 < \beta \leq 1$
Multiplied Kernel (Boolchandani & Sahula, 2011)	$K(X_i, X_j) = \alpha * K(X_i, X_j), \alpha > 0$
Log Kernel (Boolchandani & Sahula, 2011)	$K(X_i, X_j) = log(1+ \lVert X_i - X_j \rVert^\beta), 0 < \beta \leq 1$

2.2 Performance Measurements

The appropriate measurement may accurately reflect the performance of the kernel algorithms while an inappropriate one might mislead us. All the measurements are associated with the confusion matrix defined in (Tang & Zhang, 2009). TP and FP represent respectively the true and false positive class, while TN and FN stand respectively for the true and false negative class (Tang & Zhang, 2009). In Table 2, we provide the different performance metrics that are employed in the experimental task of data classification (Veropoulos, et al.1999; Tang Zhang, 2009; Boolchandani Sahula, 2011).

Table 2. Performance measurements

Accuracy	**Sensitive**	**Specificity**	**Precision**	**Recall**
$\frac{TP+TN}{TP+FP+FN+TN}$	$\frac{TP}{TP+FN}$	$\frac{TN}{TN+FP}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$
	G-means	**F-measure**		
	$\sqrt{Sensitive * Specificity}$	$\frac{2*Precision*Recall}{Precision+Recall}$		

In the particular context of imbalanced data, previous research papers reached an agreement that Accuracy is not a good metric to utilize since the influences of the two classes on the classification results are unequal (Haibo Garcia, 2009). This is due to the imbalanced numbers of instances. On the contrary, G-means and F-measure have been adopted as the experimental metric for class imbalance (Koknar-Tezel Latecki, 2009; Imam, et al.,2006). Moreover, which metric should be applied to the experiment depends also on the classification purpose and the domain of the benchmark datasets. For example, if we only care about one of the classes like the minority class, Sensitive is the most appropriate metric. If both classes are equally important, G-means is valuable as the classification metric.

3. Related Works

Imbalanced data means positive instances greatly outnumber the negative ones, or vice versa (Cao, et al.,2013). Negative instances often represent the majority (normal) class, and positive ones the minority (abnormal) class. According to (Akbani, et a., 2004), SVM performs not too badly for moderately imbalanced classes when compared to other machine learning techniques, but fails on highly imbalanced data. In the latter scenario, SVM will often classifies all instances into the majority class. Sampling techniques have been proposed to overcome the imbalanced learning problem, and which consist of under-sampling the original majority class and/or over-sampling the minority class to build a less imbalanced data (Akbani, et al., 2004; Haibo Garcia, 2009; Koknar-Tezel Latecki, 2009). In addition, sampling methods have been proved to improve the classifier performance (Haibo Garcia, 2009; Koknar-Tezel Latecki, 2009). For example, in (Japkowicz, 2000), the author examined the dual techniques of under-sampling and over-sampling and the empirical analysis demonstrated the efficiency of both methods. To make the class distribution more even, under-sampling is randomly or purposely performed to remove instances from the majority class. Under-sampling allows to reduce the size of the training sample but at the cost of deleting possibly useful data. Contrarily, over-sampling adds randomly or purposely instances in the minority class with the same objective of making a dataset less imbalanced. This technique maintains the data integrity but sacrifices the execution time because of the increase of the size of the training sample.

Many extended the classification algorithms based on the conventional under-sampling and over-sampling techniques. For instance, (Chawla, et al., 2002) presented SMOTE, a well-known over-sampling method, that generates synthetic points in the majority class in feature space. SMOTE also employs the under-sampling technique to randomly remove instances from the majority class until the proportion of minority and majority classes reach a certain level. SMOTE has been proved to improve the learning of the rare event. Indeed in (Chawla, et al., 2002), the experiments demonstrated that the proposed approach outperformed other sampling techniques in terms of Sensitivity. After that, several extensions of SMOTE have been proposed. In (Akbani, et a., 2004), the authors introduced a technique called SDC that combines a certain variant of SMOTE with several error cost algorithms. The experiments showed that SDC performed better than SMOTE in all of the ten datasets. Moreover, the SDC method outperformed other techniques such as SVM and DEC in seven out of ten datasets. In 2009, (Tang & Zhang, 2009) defined a novel algorithm, called GSVM-RU, that combines granular SVM with the under-sampling method. Furthermore, the computational cost of GSVM-RU is vastly decreased by using much less support vectors. The experiments exposed that the effectiveness of the novel method exceeded the previous best results on 12 out of 18 datasets and outperformed SVM-WEIGHT, SVM-SMOTE and SVM-RANDU (Tang & Zhang, 2009). Nevertheless, they still declared that SVM-WEIGHT should be preferentially selected for small scale imbalanced datasets. Later, (He Ghodsi, 2010) included constraints for the slack variables in the soft margin of SVM, and then optimized the objective function. Through experiments, (He Ghodsi, 2010) claimed that their method improved the performance compared to the standard SVM, but no significant improvement can be seen in these experiments.

In a previous work (Huang, et al., 2003), the authors carried out an empirical comparison of several classifiers, SVM, Naive Bayes, C4.4 and C4.5, based on two metrics, Accuracy and AUC of ROC. The classification experiments, which are performed on thirteen UCI datasets, showed that the four learning techniques have similar Accuracy. Regarding the second metric, C4.4, SVM and Naive Bayes reached similar AUC, and all significantly outperformed C4.5. Another paper (Al-Shahi, et al., 2005) demonstrated that the SVM classifier outperforms both Naive Bayes and decision trees classifiers in a particular data imbalanced problem, which is predicting protein function directly from amino acid sequence features. Before running the three classifiers, first the feature subset selection is applied and the majority category is fully under-sampled. In (Chen, 2009), the author compared three classification methods, Naive Bayes, decision tree and neural network, on rebalanced data. The experiments concluded that data re-sampling techniques remarkably increase the classification performance for the decision tree and neural network classifiers but not for Naive Bayes classifier. In particular for the decision tree, the Accuracy is significantly improved when over-sampling with SMOTE and randomly under-sampling. More recently, (Sobran, et al., 2013) compared experimentally the standard Naive Bayes with several other machine learning methods, like kNN, C4.5, SVM, and a modified version of ANN. The comparison is performed on three UCI datasets (Herbaman's Survival, German Credit and Pima Indian) by using the G-means metric. The classification results showed that Naive Bayes does not reach a good performance compared to other techniques.

4. An Analysis Approach of Imbalanced Data Classification

In this section, we expose the approach we adopted to conduct our comparative experiments, including the classification techniques and their supporting tools. In our empirical search, for each benchmark dataset, we train SVM with different kernel functions by tuning different parameters under different performance metrics. Weka is a powerful machine learning tool integrating numerous classification algorithms that can be directly applied to the selected UCI datasets. LibSVM is a library that provides the methods for the support vector classification as well as the regression and multi-class classification (Chang Lin, 2014). In (Hsu, et al., 2010), the authors stated that LibSVM is much faster than the SVM developed within the Weka tool. In order to incorporate LibSVM into Weka, we just included it into the installation path of Weka. Below, we describe the phases of our experimental approach.

4.1 Data Selection

The first step is to choose relatively new datasets from the UCI repository to make the experiments more interesting. Otherwise, it will be the same repetitive analysis work when working on the same classic data, such as Iris, Breast and Wine (Holte, 1993), since a large number of previous experiments have already tested the classification algorithms on these data. Moreover, it is useful to select datasets that have different imbalance ratios (from slightly to extremely imbalanced), different sizes (in terms of the number of instances), and different complexities (in terms of the number of attributes).

4.2 Data Preprocessing

A. Data format adaptation: first the different formats of the UCI datasets must be translated into the format that is required by the Weka interface (i.e. ARFF). One option is to copy them into an Excel document and save them as .CSV files which are then systematically transformed by Weka to .ARFF.

B. Irrelevant feature removal: here irrelevant data features must be removed as they do not contain any valuable information for the purpose of classification. Examples of these features are those that have been assigned automatically by the companies (such as customer ID). Removing these irrelevant features will eliminate the interference with other important features, and reduce the processing time of classification. Moreover, if a data feature has missing values, then delete the corresponding instances.

C. Multi-class to binary class transformation: the multi-class data are transformed into a two-class problem by merging some classes into one class.

D. Data Sampling: we need to sample the datasets for whose the classification results are unsatisfactory under any conditions. Weka employs the dual sampling techniques, under-sampling and over-sampling, to benefit from both of them. The data sampling is based on two parameters: bias and percentage. The larger the bias is, the more balanced the data will be changed to. Percentage means the total number of instances compared to the one in the original dataset. In all the experiments, we set bias to 0.5 and percentage to 100 (i.e. the total instances is the same as in the original data).

4.3 Data Classification

A. Measurement selection: G-means has been utilized as the performance metric for the classification problem of class imbalance (Imam, et al., 2006). We also pay attention to Sensitive since it provides the accuracy for the minority class, which we are most concerned with when classifying imbalanced data. In many real-life scenarios, the rare case is often of interest. For example, in online auctions, we consider that misclassifying fraudulent as normal behaviour is much worse than misclassifying a normal behaviour as fraudulent. The consequence for the latter one might be sending a warning to the bidder and verifying further his behaviour to confirm or reject the suspicion. Nevertheless, the former will result in money loss for honest bidders. Moreover, to make the comparative experiments more comprehensive and have a better idea about the performance of each classifier, we employ in total three metrics: Sensitive, Specificity (measures the accuracy of the majority samples), G-means (measures the accuracy of both classes). In addition, we compute the processing time for each classifier on each dataset.

B. Classification with SVM: we first choose several kernel functions from Table 1, and then apply them to each dataset under different parameter settings as well as different performance metrics. We train SVM with three types of kernels: Linear, Polynomial, and non-linear RBF, as they are widely used in practice. We have in total three tuneable parameters: the penalty C, parameter γ, and the K-fold cross validation. The purpose here is to examine the influence of each parameter on each kernel.

The parameter C, called penalty, regulates the trade-off between training error minimization and margin maximization Mulier 2007). When C is too big, overfitting may occur, and when it is too small, underfitting may occur. The K-fold cross-validation consists of splitting a dataset into K subsets, (K-1) as the training sample and 1 as a testing sample. The cross-validation method may stop the overfitting issue. We vary K from a small value to a larger one until the classification results reach a stable and relatively good state. However, the time complexity of classification is increased by K, so we always try to find a satisfiable but as small as possible value of K.

To attain the best results for SVM, we need to run numerous experiments for each dataset by varying the three parameters for each kernel and under each performance measurement. For each data, the learned model is the kernel function and parameter setting that achieved the best performance. This model will be used in the future to predict the category of a new instance.

C. Classification with Naive Bayes and J48: In addition to SVM, we apply the other two widely used classifiers on each dataset. Naive Bayes is a simple but important probabilistic classifier based on the Bayes theory and independence assumption (Collins, 2014). J48 or C4.5 is a decision tree classifier proposed by Quinlan (Quinlan, 1993). To achieve the best results for these two classifiers, we need to run several experiments by changing the parameter K on each dataset under each measurement. These two classifiers utilize the default values set by Weka.

4.4 Classifier Comparison

Here, we discuss the best classification results achieved by the three machine learning tools, SVM, Naive Bayes and J48 (C4.5), on each dataset under each performance metric: Sensitive, Specificity, G-means and Processing Time. We also compare the produced results to determine the best classifier for each dataset.

5. Empirical Analysis and Comparison

This section exposes the experimental results of classifying new datasets with the three classifiers. We repeat the same experiments on all the datasets by testing different kernels and parameters. We only present the most significant classification results. The experiments are conducted on Windows XP with Intel Core 2 Duo CPU T6400 2.00 GHz, and the figures are generated by using Matlab 2012a.

5.1 Data Selection

With the purpose of making the experiments more interesting, from diverse areas, we collect five new datasets from the UCI repository. Table 3 presents the major information of these datasets. To the best of our knowledge, there are no previous works that showed the performance of the classification algorithms on these data. The size of these five pieces of data varies from a small scale (Fertility with 100) to a large scale (Bank Marketing with 4521). Maj. and Min. represent the number of instances in the majority and minority class respectively, and Att. the number of attributes. The imbalance ratio varies from a slight imbalance (User Knowledge Modelling with 151:107) to an extreme imbalance (Seismic Bumps with 2414: 170).

Table 3. Benchmark datasets from UCI

Data	Ins.	Maj.	Min.	Att.	Year	Area
Fertility	100	88	12	10	2013	Biomedical
UserKnowledgeModeling	258	151	107	6	2013	Education
Vertebral Column	310	210	100	6	2011	Biomedical
Seismic Bumps	2564	2414	170	19	2013	Coal mining
Bank Marketing	4521	4000	521	17	2012	Business

5.2 Data Preprocessing

Below we summarize the preprocessing tasks for each piece of data. The first step is to transform the dataset formats to the one required by the Weka software. The User Knowledge Modelling data divides the student knowledge levels into four categories: *verylow*, *low*, *middle* and *high*. In order to transform it into a binary classification problem, we classify *verylow* and *low* levels into the class *fail*, *middle* and *high* into the class *pass*. In the dataset of Bank Marketing, we remove four irrelevant features: contact type, last contact day of the month, last contact month of year, and last contact duration. In our opinion, these features do not contain useful information for the classification.

We found that it is difficult for the three classifiers to process highly imbalanced datasets with satisfactory results. For instance, the three selected kernels perform poorly in terms of Sensitive in the original data of Seismic Bumps and Bank Marketing. As a result, we sample these two data to make them less imbalanced by setting bias to 0.5 and percentage to 100. Now Seismic Bumps data has a lower imbalance ratio of 1819:745, and Bank Marketing of 3097:1421. We may note that Fertility and VertebralColumn data do not require any preprocessing tasks.

5.3 Data Classification

We train SVM with Linear, Polynomial with degree 2, and Radial Basis Function (RBF). We keep adjusting the three parameters K, C and γ until we reach satisfactory results for SVM.

5.3.1 Fertility

Table 4 indicates that Naive Bayes and J48 have difficulty addressing the minority instances for Fertility: the performance of Sensitive is only 0.083 and 0.167 respectively. On the other hand, SVM proves outstanding when classifying the minority class, reaching 1 under the setting of $K = 5$ and $C = 0.01$ with the linear kernel function, though it sacrifices the performance for the majority class. We conduct several trials for SVM and unfortunately we could not find a combination of a kernel function and parameter setting that classifies both classes with an acceptable result for the G-means metric. Actually, if we change the parameter setting to improve the Specificity, the Sensitive is considerably decreased and vice versa. In fact, the trade-off results for both Sensitive and Specificity are only 0.417 and 0.534 respectively under the parameter setting of $K = 2$, $C = 0.1$ with Linear function.

Table 4. Classifier comparison for fertility

	SVM (K=5, C=0.01, Linear)	SVM (K=5, C=0.5, Linear)	Naive Bayes (K=5)	J48 (K=5)
Sensitive	1	0	0.083	0.167
Specificity	0	1	0.966	0.932
G-means	0	0	0.283	0.395

5.3.2 User Knowledge Modeling

For the evaluation of the student knowledge level, this time we care about both “fail“ and “pass“ classes, so that the measurement of G-means is appropriate for this data. Table 5 shows the best performance achieved by the three kernel functions by varying the three parameters, which are all satisfactory. Since it is a small-scale dataset, the building time of the learning model of the three kernels is very fast, less than 0.1 seconds.

Table 6 compares the best overall performance of G-means achieved by SVM, Naive Bayes and J48. Apparently, SVM outperforms the other two classifiers. However, the performances of other classifiers are also acceptable. It turns out that since this data is slightly imbalanced, it can be well handled by the different classifiers.

Table 5. Kernel evaluation for user knowledge

	Linear	Polynomial	RBF
	K=10, C=5	K=5, γ=2, C=3	K=10, γ=0.5, C=0.5
Sensitive	1	0.991	1
Specificity	0.967	0.967	0.954
G-means	0.983	0.978	0.976

Table 6. Classifier comparison for user knowledge

	SVM	Naive Bayes	J48
G-means	0.983	0.956	0.974

5.3.3 Vertebral Column

We have conducted many experiments on this dataset and found out that Linear and Polynomial functions are both very time consuming with the average building model time of 29s and 91s respectively. Meanwhile, RBF performs efficiently, spending less than 0.1s to generate the learning model. Hence, RBF is the most appropriate kernel for this dataset. Table 7 presents the classification results when varying the penalty C under $K = 5$ and $\gamma = 0.1$ with RBF. So, we can see that when $C <= 0.39$, Sensitive reaches its highest point and stays at this point no matter how small C would be. Moreover, from the value of 0.39, Sensitive decreases with the increase of C until it reaches 0. Figure 2 illustrates the tendencies of Sensitive and Specificity with the change of penalty C. These two metrics are always on the contrary, and if one of them increases, the other decreases. If we vary γ, the pattern is observed.

Table 8 compares the classification results of SVM, Naive Bayes and J48. We can see that SVM can accurately classify minority or majority instances under a certain parameter setting, but cannot correctly classify both of them at the same time. For example, the Sensitive reaches 1 by SVM under $K = 5$, $C = 0.3$, $\gamma = 0.1$, meanwhile the Specificity is 0. The best overall performance of G-means is obtained by SVM with $K = 5$, $\gamma = 0.1$ and $C = 0.47$. For this measurement, SVM is not as good as the other two classifiers. Naive Bayes has the best overall accuracy. In conclusion, if we consider the overall classification metric, Naive Bayes is the best choice for this data. If we only care about the accuracy of one of the classes, SVM is the most accurate. From our perspective, SVM efficiently deals with the minority class at the cost of sacrificing the performance for majority class in this data. So far we could not found any satisfactory trade-off setting for this dataset.

5.3.4 Rebalanced Seismic Bumps

For this large and complex dataset, Linear and Polynomial kernels perform miserably due to the extremely high computing time, both spending more than 600s to build the classifier model and almost triple time for the cross validation. Meanwhile, RBF just takes around 5s to build the model and 5s to 20s for the cross validation, depending on the K value.

Table 7. Varying penalty C for RBF-based SVM

Penalty	0.39	0.40	0.45	0.46	0.47	0.48
Sensitive	1	0.99	0.97	0.96	0.25	0
Specificity	0	0	0.076	0.19	0.986	1
G-means	0	0	0.272	0.43	0.50	0

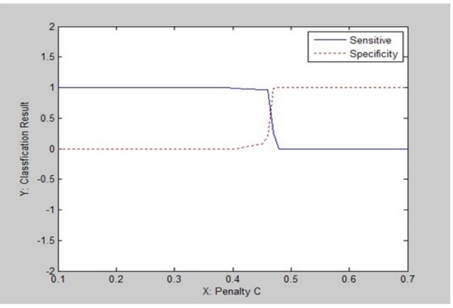

Figure 2. Tendency of sensitive and specificity with Penalty C

Table 8. Classifier comparison for vertebral column

	SVM (K=5, C=0.3, γ=0.1)	SVM (K=5, C=0.5, γ=0.1)	SVM (K=5, C=0.47, γ=0.1)	Naive Bayes (K=5)	J48 (K=10)
Sensitive	1	0	0.986	0.88	0.64
Specificity	0	1	0.25	0.71	0.9
G-means	0	0	0.50	0.79	0.76

In Table 9, the classification is conducted under 5-fold cross validation, and the best results are obtained by RBF with 0.966, 1, and 0.98 for Sensitive, Specificity and G-means respectively, regardless of the values of γ. In fact, the parameter γ has no influence on the final results. Moreover, the interesting thing is that all the results reach the highest performance under the penalty C of 0.77.

Further, Table 10 clearly shows that the results are only influenced by K because we can always find a penalty C for different K-fold cross-validation to achieve the highest Sensitive. Unlike the situation in Vertebral Column and Fertility datasets, SVM can reach the highest Sensitive and Specificity at the same time. For example, when $K = 10$, it can reach the highest result of 0.983 and 1 for minority and majority class respectively under $C = 0.78$. Besides, we find out that when K is greater than 10, the highest Sensitive will stay at 0.983 and no longer changes with the growth of K. Obviously, SVM outperforms Naive Bayes and J48 under each measurement, while Naive Bayes is the worst in terms of addressing the imbalanced data with poor Sensitive of 0.55.

Table 9. Parameter setting for best RBF performance

γ	C	Sensitive	Specificity	G-means
0.1	0.77	0.966	1	0.98
1	0.77	0.966	1	0.98
10	0.77	0.966	1	0.98
100	0.77	0.966	1	0.98

Table 10. Classifier comparison for seismic bumps

	Sensitive	Specificity	G-means	K	C
SVM (RBF)	0.887	1	0.94	2	0.74
	0.966	1	0.98	5	0.77
	0.983	1	0.99	10	0.78
	0.983	1	0.99	20	0.78
NaiveBayes	0.55	0.849	0.68	5	-
J48	0.942	0.928	0.93	10	-

5.3.5 Rebalanced Bank Marketing

Similar to the Seismic Bumps data, Linear and Polynomial are very time consuming since this dataset is large and complex. From Table 11, we can see that SVM has the outstanding performance for both Sensitive and Specificity on the sampled Bank Marketing data. When K=10, SVM performs better when compared to K=2 and K=5. Actually, SVM achieves 1 and 0.919 for minority and majority classes respectively. It is probable that the training set is not enough to train the best model when K is less than 10. In essence, the overall performance achieved by J48 is satisfactory while Naive Bayes shows a deficiency for classifying minority instances.

Table 11. Classifier comparison for bank marketing

	Sensitive	Specificity	G-means	K	C
SVM (RBF)	0.749	1	0.87	2	0.3
	0.89	1	0.94	5	0.21
	1	0.919	0.96	10	0.21
Naive Bayes	0.638	0.848	0.74	5	-
J48	0.904	0.927	0.92	25	-

5.4 Classifier Comparison

Figure 3 presents the evaluation of the three classifiers in terms of Sensitive on each of the four datasets. We may note that User Knowledge Modeling is only learned by the G-means metric since it is a slightly imbalanced dataset, so the two classes are equally important. We can see that SVM outperforms Naive Bayes and J48 in all the four data. In particular, regarding data Fertility, both Naive Bayes and J48 show their huge weakness in classifying correctly the minority instances. Naive Bayes performance is unsatisfactory for those large datasets such as Bank Marketing and Seismic Bumps. It is also difficult for J48 to classify minority instances for Vertebral Column.

In Figure 4, for classifying both classes at the same time, SVM outperforms the other two classifiers for three datasets excluding Fertility and Vertebral Column. Both J48 and Naive Bayes show a better performance in these two small-scale data. We may note that Fertility and Vertebral Column seem the most difficult data to address because none of the three classifiers reached more than 90% for the G-means metric: less than 50% for Fertility, and less than 80% for Vertebral Column. In contrast, User Knowledge is the easiest data to be handled by the three classifiers with at least 95% for G-means. We may note that for large and complex datasets, SVM is much better that the two other classifiers.

Overall, Vertebral Column and Fertility, thought small scale data, are the most difficult to be correctly classified. It is not as expected as most researchers consider that smaller the scale is, the easier the classification will be. Nonetheless, the penalty C considerably influences a certain data but has almost no influence on another data, and the same can be concluded regarding the other tuning parameters.

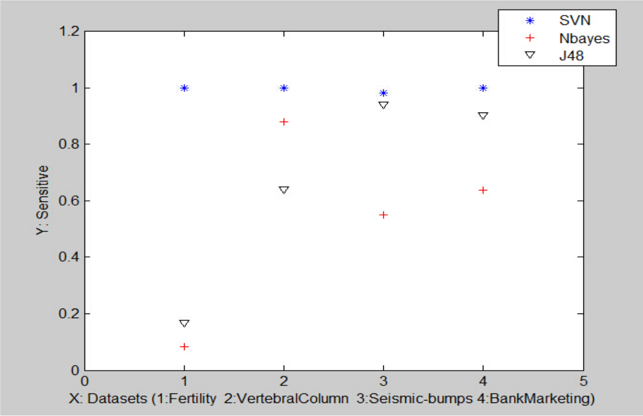

Figure 3. Comparison of sensitive

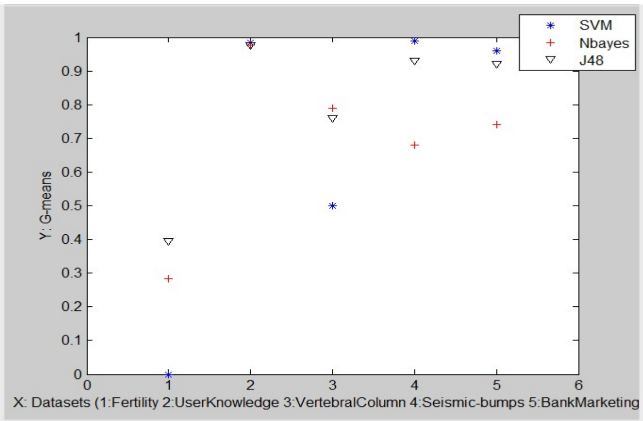

Figure 4. Comparison of G-means

6. Conclusion and Future Work

In this paper, we conducted various experiments on five new UCI imbalanced datasets by applying three popular classifiers, SVM, Naive Bayes and Decision tree (C4.5). Thanks to our empirical search, we determined the learning model for each dataset. The data classification with SVM is considerably impacted by the kernel functions, penalty C, and cross validation. Actually not all the parameters play a major role to influence the classification performance. According to the experiments, SVM outperforms Naive Bayes and Decision tree (C4.5) in terms of correctly classifying the minority instances (or the majority instances) for all the benchmark datasets. However, in only two small-scale datasets (Vertebral Column and Fertility) SVM shows its deficiency for correctly classifying both classes at the same time, attaining high Sensitive at the cost of sacrificing the Specificity and vice versa.

Interesting research works have been carried out in order to increase the learning rate of the minority events by extending and adapting the SVM classifier (Koknar-Tezel Latecki, 2009; Imam, et al.,2006; Haibo Garcia, 2009). In online auctions, we are interested in classifying fraudulent users who represent the minority category. Auction fraud activities, representing the largest part of all Internet frauds, are on the rise because auctions give many opportunities for conducting misbehaviour (Mamum Sadaoui, 2013). These frauds are very important to detect as they lead to money loss for honest bidders. For instance, in 2008, around $250 million may have been lost to shills in the auction house of e-Bay (Cohen, P., 2009). Hence, we would like to investigate which extension of SVM is the most suitable specifically in the context of online auctions.

References

Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. In *the 15th European Conference on Machine Learning*, vol. 3201, pp. 39-50. Springer Berlin Heidelberg.

Al-Shahib, A., Breitling, R., & Gilbert, D. (2005). Feature Selection and the Class Imbalance Problem in Predicting Protein Function from Sequence, *Applied Bioinformatics, 4*(1), 95-203.

Boolchandani, D., & Sahula, V. (2011). Exploring efficient kernel functions for support vector machine based feasibility models for analog circuits. *International Journal of Design, Analysis and Tools for Circuits and Systems,* 1, 117-128. Springer US.

Cao, P., Zhao, D., & Zaiane, O. (2013). An Optimized Cost-Sensitive SVM for Imbalanced Data Learning. *Advances in Knowledge Discovery in Computer Science, 7819*, 280-292. Springer Berlin Heidelberg.

Chang, C. C., & Lin, C. J. (2014). *LIBSVM - A Library for Support Vector Machines*. Retrieved from http://www.csie.ntu.edu.tw_ cjlin_libsvm/

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research, 16*, 321–357. AI Access Foundation, USA.

Chen, Y. (2009). Learning Classifiers from Imbalanced, Only Positive and Unlabelled Data Sets. Project Report for UC San Diego Data Mining Contest, Department of Computer Science Iowa State University.

Cherkassky, V., & Mulier, F. M. (2007). *Learning from Data: Concepts, Theory, and Methods*. John Wiley and Sons, IEEE Press.

Cohen, P. (2009). Shill Bidding on eBay: A Case Study or the facilitating and concealing of frauds by eBay. In AuctionBytes Forum, Online Auction News Forum, 2009.

Collins, M. (2014). *The Naive Bayes Model, Maximun-Likehood Estimation and the EM algorithm*. Retrieved from http://www.cs.columbia.edu_mcollins/em.pdf

Cortes, C., & Vapnik, V. (2014). *Support-Vector Networks*. Retrieved from http://homepages.rpi.edu/ bennek/class/_mmld/papers/svn.pdf

Haibo, H., & Garcia, E. A. (2009). Learning from imbalanced data. In *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1263-1284. IEEE Educational Activities Department Piscataway, NJ, USA.

He, H., & Ghodsi, A. (2010). Rare class classification by support vector machine. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, pp. 548-551. IEEE.

Holte, R. C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. In *Machine Learning, 11*, 63-91. Kluwer Academic Publishers-Plenum Publishers.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2010). *A practical guide to support vector classification*. Retrieved from http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf

Huang, J., Lu, J., & Ling, C. (2003). Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy. In *Third IEEE International Conference on Data Mining (ICDM03)*, pp. 1-4.

Imam, T., Ting, K., & Kamruzzaman, J. (2006). Z-SVM: An SVM for Improved Classification of Imbalanced Data. *AI 2006: Advances in Artificial Intelligence, 4304*, 264-273. Springer Berlin Heidelberg.

Japkowicz, N. (2000). The class problem: significance and strategies. In *2000 International Conference on Artificial Intelligence (ICAI)*, pp. 111-117.

Koknar-Tezel, S., & Latecki, L. J. (2009). Improving SVM Classification on Imbalanced Data Sets in Distance Spaces. In *Ninth IEEE International Conference on Data Mining*, vol. 17, pp. 259-267. IEEE.

Mamum, K., & Sadaoui, S. (2013). Combating Shill Bidding in Online Auctions. In *International Conference on Information Society, i-society*, pp. 170-176. IEEE press.

Muller, K., Mika, S., Ratsch, G., & Tsuda, K. (2002). An introduction to kernel-based learning algorithms. In *Neural Networks, 12*, 181-201. IEEE.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc.

Sobran, N. M. M., Ahmad, A., & Ibrahim, Z. (2013). Classification of imbalanced dataset using conventional naive bayes classifier. In *International Conference on Artificial Intelligence in Computer Science and ICT*, pp. 35-42.

Tang, Y., & Zhang, Y. (2009). SVM modeling for highly imbalanced classification. In *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics - Special Issue on Human Computing*, vol. 39, (pp. 281-288). IEEE Press.

UCI (2013). Centre for machine learning and intelligent systems. Retrieved from http://archive.ics.uci.edu/ml

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley Interscience.

Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the Sensitivity of Support Vector Machines. In *International Joint Conference on Artificial Intelligence*, pp. 55-60.

Copyrights