A divide and conquer approach for imbalanced multi-class classification and its application to medical decision making

Hu Li
College of Computer, National University of Defense Technology, Changsha, China

Abstract: Many real world data contains more than two categories and the number of instances in each category differs greatly. Such as in medical diagnostic data, there may be several types of cancer and each with tens instances, but contains even more normal instances. Similarly, there may be very few abnormal samples in pharmaceutical test but which may cause great harm. Classification of such type of data is often summarized as imbalanced multi-class classification. Most existing researches study multi-class classification and imbalanced data classification separately, few study in a combination way, in particular for medical diagnosis data classification. In the context of medical diagnosis and pharmaceutical test, in this paper, we propose a divide and conquer approach to partition multi-class data and a self-adaptive data resample method for imbalanced data. The proposed methods are tested on 23 UCI datasets in medical, pharmaceutical and other fields. Experiment results show that the proposed methods outperform other compared methods, in particular on those medical and pharmaceutical dataset.

Keywords: Multi-class, imbalance, medical diagnostic; classification.

INTRODUCTION

Machine learning and data mining methods are often used in medical decision making and pharmaceutical test. Traditional method default assumes that the distribution of the trainset is uniform. However, many practical problems are not consistent with this hypothesis, such as in medical diagnosis (Luis and Jesus, 2006; Sanz *et al*., 2014), pharmaceutical test (Eitrich *et al*., 2007; Li *et al*., 2014) and fraud detection (Sahin *et al*., 2013). Meanwhile, many real world datasets contain more than two categories, such as the Thyroid disease dataset, which includes 6666 normal instances, 166 type-I cancer instances and 368 type-II cancer instances. Similarly, chemicals can be classified into four categories, namely categories I, II, III and IV, based on EPA. Imbalanced multi-class dataset not only contains more categories but also may include imbalance within minority classes, such as type-I cancer instances is much less than type-II cancer instances. All those problems further increase the difficulty of imbalanced multi-class classification (IMCC). Therefore, the study is full of challenges but with great practical significance.

Solutions for IMCC problem can be divided into algorithm level and data level. The first type method is designed specifically for IMCC problem, such as imbalanced multi-class SVM (Phoungphol *et al*., 2012) and cost sensitive ensemble learning (Sun *et al*., 2006). Some researches make use of K-means clustering algorithm, such as KSMOTE (Prachuabsupakij and Soonthornphisaj, 2012) and BCT-OB (Athimethphat and Lerteerawong, 2012). Meanwhile, multi-class decision tree is also used as to cope with IMCC problem, such as MC-HDDTs (Hoens *et al*., 2012).

The second type method tries to divide IMCC problem into a series of two-class classification problems and make use of existing two-class algorithms. Those methods mostly use divide and conquer method but with different partition strategies. There are mainly four partition strategies and many other researches are based on them. The first one is One-Versus-All (OVA) (Rifkin and Klautau, 2004). Suppose trainset D contains K categories. OVA take one category as positive and all other $K-1$ categories as negative each time and a two-class sub-classifier is trained. Totally, there are K sub-classifiers trained. The second one is One-Versus-One (OVO) (Hastie and Tibshirani, 1998). OVO take one category as positive and another one as negative each time and train a two-class sub-classifier. There are $K(K-1)/2$ sub-classifiers trained totally. The third one is All-and-One (A&O) (Garcia-Pedrajas and Ortiz-Boyer, 2006). A&O make use of OVA and OVO together and $K(K+1)/2$ sub-classifiers need to be trained. For each test instance, A&O first use OVA sub-classifiers to find top two most likely categories and then use corresponding OVO sub-classifier to determine the final result. The last one is One-Against-Higher-Order (OAHO) (Murphey *et al*., 2007). OAHO first sort all categories by number of instances in descending order, after which the current category is selected as positive and all other subsequent categories as negative. Finally, $K-1$ sub-classifiers will be trained. Many following researches are based on these four partition strategies. Such as combination of OVO, SMOTE and Linguistic Fuzzy Rule (Fernández *et al*., 2010), OVA with Data Balancing (Jeatrakul and Wong,

**Corresponding author:* e-mail: lihu@nudt.edu.cn

2012), Multi-IM (Ghanem *et al*., 2010) which combined A&O and PRMs-IM (Ghanem *et al*., 2008).

We have briefly introduced several related methods at algorithm level and data level respectively. But it should be noted that those two level methods are often used together, such as in BCT-OB (Athimethphat and Lerteerawong, 2012) and MC-HDDTs (Hoens *et al*., 2012). Comparison on 24 different types of datasets showed that OVO combined with data resample or with cost sensitive algorithm can usually achieve better results (Fernández *et al*., 2013).

Aforementioned methods focus more on multi-class than on class imbalance, which is also important in IMCC. Existing methods for imbalanced data classification can also be divided into algorithm level and data level. Here, we mainly focus on data level method for its simplicity. Data level methods try to balance trainset through resampling and can be further divided into over-sample on minority and under-sample on majority. The simplest way is random copy minority instances or deletes majority instances, but it is likely to cause over fitting problem or loss of useful information. Thus, various heuristics resample methods were proposed to cope with those problems. The most widely used is SMOTE (Chawla *et al*., 2002) proposed in 2002. Many follow-up works are based on SMOTE, such as Borderline-SMOTE (Han *et al*., 2005) which suppose that only those instances near the borderline can help improve classification results. Similarly, Safe-level-SMOTE (Bunkhumpornpat *et al*., 2009) assign a safe level to each minority instance before synthesizing new instance. DBSMOTE (Bunkhumpornpat *et al*., 2011) synthesize new instances based on density. ADASYN (He *et al*., 2008) take difficulty of classify minority instances into consideration.

By combing existing researches it can found that for IMCC problem, divide and conquer method is more concise and usually have better results. Therefore, based on actual demand of decision making in medical diagnosis and pharmaceutical test, in this paper, we propose an improved divide and conquer approach and a self-adaptive data resample methods to cope with IMCC problem. In the following, we first formalize IMCC problem and then propose the divide and conquer framework for IMCC and describe it in detail. Then, self-adaptive imbalanced data resample method is presented. After that, experiments and discussion of the results are provided, followed by conclusion and outlook.

PROBLEM FORMALIZATION

Assuming dataset $X=\{x_1,x_2,\ldots..,x_n\}$contains n instances, in which $x_i \in R^m$ is the *i*-th instance in X and x_i contains m attributes. Assuming class label set $Y=\{c_1,c_2,\ldots..,c_K\}$, in which K is the number of categories and K>2. Then, for trainset $D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$, in which $x_i \in X$, $y_i \in Y$. Learning task is to train one or set of classifiers f which can correctly map X to Y, i.e. $f: X \to Y$. For new test instance x, f can accurately predict its class label, i.e. f(x)=y'=y, in which y' is the predicted class label and y is the true class label. When K=2, multi-class classification problem degenerates into two-class classification problem.

For imbalanced two-class dataset, its imbalance ratio (IR) can be defined as $IR = N_{maj}/N_{min}$, in which N_{maj} is the number of majority instances and N_{min} is the number of minority instances. But in IMCC problem, calculation of IR is more complicated. Usually, there are two different calculation ways, corresponding to OVO and OVA. (1) Firstly, calculate IR between each two categories. For category i and j , assuming corresponding number of instances are N_i and N_j respectively, then, $IR(i,j) = \max\{N_i, N_j\}/\min\{N_i, N_j\}$ i,j=1,⋯,K $i \neq j$. Finally, take the maximum value as imbalance ratio of the dataset, i.e. $IR = \max\{IR(i,j) | i,j = 1,\cdots,K \; i \neq j\}$; (2) Firstly, calculate IR of each category relative to all other categories, i.e. $IR(i) = \max\{N_i, n - N_i\}/\min\{N_i, n - N_i\}$ i=1,⋯,K . Finally, take the maximum value as imbalance ratio of the dataset, i.e. $IR = \max IR(i) \;\; i = 1,\cdots,K$.

IMCC FRAMEWORK

As mentioned previously, we take divide and conquer method in this paper to deal with IMCC problem. Main idea of our proposed method is: (1) at first, we use OVA to ensure that all sub-classifiers are trained on entire trainset. In order to ensure the trainset is original, we do not process it at this stage; (2) then, use OVO to train sub-classifiers in a more fine-grained level. At this stage, each subset will be checked to find whether need resample or not, and all those subsets with imbalance ratio higher than threshold will be resampled; (3) in A&O, only two classes with the highest and second highest value will be selected and inputted to corresponding OVO sub-classifier. In this paper, we select all classes with value higher than predefined threshold but not limit candidate class number to two. In this way, more instances can be recalled.

For trainset D , we first use OVA to divide D into K subsets $D_1^{OVA}, D_2^{OVA}, \cdots, D_K^{OVA}$, in which D_i^{OVA} take category i as positive and all remaining categories as negative. After that, train sub-classifiers $f_1^{OVA}, f_2^{OVA}, \cdots, f_K^{OVA}$ on each subset. Depending on learning algorithm, the output of each sub-classifier can be 1 or 0, indicates test instance belongs to given class or not respectively, such as decision tree. The output can also be the probability of test instance belonging to given class, such as Naïve Bayes. Since discrete and numeric output corresponding to different process methods, we describe our framework for those two kinds of output separately.

In case of discrete output, our proposed framework is described in Algorithm 1. For test instance x, outputs from each sub-classifier can be divided into three possible cases. (1) Only one output is 1 and all other sub-classifiers' output is 0. In this case, the classification result is unique and the class label corresponding to i is selected as final output; (2) all outputs are 0. In this case, the test instance x is predicted to none of existing categories. All sub-classifiers take the test instance x as negative because of imbalance within each subset. It means each category is taken as minority when compared with all other categories. In this case, we resample all those subsets with imbalance ratio higher than threshold and resample method will be presented in next section; (3) multiple outputs are 1. In this case, we use OVO to divide those corresponding categories. Thereafter, calculate the imbalance ratio within those subsets and do resampling when necessary. Then, new sub-classifiers will be trained, For test instance x, the final class label is determined by majority voting.

Algorithm 1. IMCC Framework-discrete output

Input: Trains et $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$
Test instance x
Imbalance ratio threshold T
Sub-classifier output threshold λ

Output: Predicted class label $label(x)$

1. Divide D into K subsets $D_1^{OVA},D_2^{OVA},\cdots,D_K^{OVA}$
2. Train sub-classifier on each subset and get $f_1^{OVA}, f_2^{OVA},\cdots, f_K^{OVA}$
3. Input x to each sub-classifier and statistic all outputs
4. If $f_i^{OVA}(x)=1$ and $f_j^{OVA}(x)=0\ j\neq i$, then $label(x)=c_i$
5. If $f_i^{OVA}(x)=0\ \ i=1,2,\cdots,K$, then
6. Resample on $D_1^{OVA},D_2^{OVA},\cdots,D_K^{OVA}$
7. Train new sub-classifiers on resampled subsets and test x again
8. If $2\le\left(\sum_{i=1}^{K} f_i^{OVA}(x)=K_1\right)\le K$, then
9. Use OVO to divide all corresponding categories into $m_1=K_1(K_1-1)/2$ subsets $D_1^{OVO},D_2^{OVO},\cdots,D_{m_1}^{OVO}$
10. If imbalance ratio within $D_i^{OVO}\ (i=1,\cdots,m_1)$ higher than T, then resample it
11. Train sub-classifiers on resampled subset and get $f_1^{OVO}, f_2^{OVO},\cdots, f_{m_1}^{OVO}$
12. Input x to all OVO sub-classifiers and determine the final result using majority voting, i.e. $label(x)=majorityVote(x, f_1^{OVO}, f_2^{OVO},\cdots, f_{m_1}^{OVO})$

As for numeric output, we first sort all OVA sub-classifiers' output in descending order. Then, those categories corresponding to output value higher than threshold are selected as candidates. Thereafter, depending on number of candidates, different strategies are used and the procedure is detailed in Algorithm 2.

Algorithm 2. IMCC Framework-numeric output

Input: Trainset $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$
Test instance x
Imbalance ratio threshold T
Sub-classifier output threshold λ

Output: Predicted class label $label(x)$

1. Divide D into K subsets $D_1^{OVA},D_2^{OVA},\cdots,D_K^{OVA}$
2. Train sub-classifier on each subset and get $f_1^{OVA}, f_2^{OVA},\cdots, f_K^{OVA}$
3. Input x to each sub-classifier and get K numeric outputs $o_1,o_2,\cdots,o_K$
4. Sort all outputs in descending order and get $o_1^{'},o_2^{'},\cdots,o_K^{'}$
5. If $o_1^{'}\ge\lambda$ and $o_i^{'}<\lambda\ (i=2,\cdots,K)$, then $label(x)$=$labelOfOutput(o_1^{'})$
6. If $o_1^{'}\ge o_2^{'}\ge\lambda$ and $o_i^{'}<\lambda\ (i=3,\cdots,K)$, then
7. Extract all instances corresponding to $o_1^{'}$ (class s) and $o_2^{'}$ (class t), get $D_{s,t}^{OVO}$
8. If imbalance ratio on $D_{s,t}^{OVO}$ higher than T, then resample it
9. Train sub-classifier $f_{s,t}^{OVO}$ on $D_{s,t}^{OVO}$
10. Label of x will be determined by $f_{s,t}^{OVO}$, i.e. $label(x)=f_{s,t}^{OVO}(x)$
11. If $o_1^{'}\ge o_2^{'}\ge\cdots\ge o_{K_2}^{'}\ge\lambda,\ (3\le K_2\le K)$ and $o_i^{'}<\lambda,(K_2<i\le K)$, then
12. Extract all instances corresponding to $o_1^{'},o_2^{'},\cdots,o_{K_2}^{'}$ and get $D^{'}$
13. Divide $D^{'}$ into $K_2(K_2-1)/2=m_2$ subsets $D_1^{'},D_2^{'},\cdots,D_{m_2}^{'}$ using OVO
14. If imbalance ration on $D_i^{'}$ higher than T, then resample it
15. Train sub-classifiers on resampled subsets and get $f_1^{OVO}, f_2^{OVO},\cdots, f_{m_2}^{OVO}$
16. Predict class label of x using majority voting, i.e. $label(x)=majorityVote(x, f_1^{OVO}, f_2^{OVO},\cdots, f_{m_2}^{OVO})$

We have presented our framework for IMCC problem above in detail. Compared with existing researches, we cope with the problem in a more fine-grained manner, more candidates are selected using different threshold, and the resampling is always done when necessary. In next section, we will describe our resample method in detail.

SELF-ADAPTIVE DATA RESAMPLING

Our self-adaptive imbalanced data resampling (SIDR) method include neighbor-based over-sample (NBOS), distance-based under-sample (DBUS) and sample ratio adjust strategy.

Neighbor-based over-sample

Our proposed neighbor-based over-sample method is described in detail in Algorithm 3. Firstly, we calculate the imbalance ratio between majority and minority. The imbalance ratio will be used later to determine how many minority instances will be synthesized. Then, for each minority instance x, calculate distance between x and other minority instances and find the nearest k neighbors. Then, for each x_i in K neighbors, calculate and find the nearest K neighbors around x_i. At this stage, the nearest K neighbors around x_i may include majority instances. After that, we statistic the ratio between majority instances and minority instances for x_i's k neighbors. The larger the ratio means more majority instances surround x_i. Therefore, probabilities of select x_i for synthesis new instances should be lower. Finally, those neighbors with high probabilities are more likely to be selected to synthesis new minority instances.

Algorithm 3. Neighbor-based Over-sample
Input: Trainset $D=\{(x_1,y_1)\cdots(x_n,y_n)\}$
Number of the nearest neighbors k
Output: Resampled balanced trainset D
1. Calculate imbalance ratio between majority and minority, i.e. $IR = N_{maj} / N_{min}$
2. For each minority instance x
3. Calculate and find the nearest k neighbors, i.e. $x_i \ i=1,\cdots,k$
4. For each instance in k neighbors, i.e. $x_i \ i=1,\cdots,k$
5. Calculate and find the k nearest neighbors, i.e. $x_{ij} \ j=1,\cdots,k$
6. Calculate the class ratio in x_{ij}'s k nearest neighbors, $r_i = \#\{x_{ij}='+'\} / \#\{x_{ij}='-'\}$
7. Sort r_i in ascending order
8. Select x_i with higher r_i as candidate for synthesis, $p(x_i) = power(random(0,1),2)\cdot k$
9. Synthesis new minority instances $x^{'} = x + (x - x_i)\times\lambda \quad \lambda \in random(0,1)$
10. Repeat step 9 according to number of minority needed
11. Combine D and all new synthesized minority instances together, get D

Distance-based under-sample

Under-sample aims at reducing majority instances, especially those "redundant" instances, and thus reducing the imbalance ratio. Existing methods, such as CNN (Hart, 1968), ENN (Wilson, 1972) and Tomek Link (Tomek, 1976), try to find "noisy" or "redundant" majority instances through iterative calculations. When the number of majority instances increase sharply, those methods also bring high computation complexity. In this paper, we present a simple distance-based under-sample method described in Algorithm 4. Firstly, we calculate the number of majority instances N need to be removed. After that, for each majority instance x, calculate the distance from itself to all other minority instances and then average the distance. Thereafter, averaged distances will be sorted in descending order and top N instances will be removed from sorted list. Finally, all remained instances will be outputted.

Algorithm 4. Distance-based Under-sample
Input: Trainset $D=\{(x_1,y_1)\cdots(x_n,y_n)\}$
Output: Resampled balanced trainset $D^{'}$
1. Calculate the number of majority instances need to be removed, i.e. $N = N_{maj} - N_{min}$
2. For each majority instance x
3. Calculate distances between x and all other minority instances
4. Get average distance d_x
5. Sort d_x in descending order, get $d_x^{'}$
6. Remove top N instances in $d_x^{'}$
7. Take remained instances as $D^{'}$

Sample ratio adjust strategy

Algorithm 5 describes a self-adaptive sample ratio adjust strategy which can dynamically calculate the number of samples need to synthesis or removed when given expected imbalance ratio. Finally, we get a balanced dataset with equal size with original one.

Algorithm 5. Self-adaptive resample
Input: Trainset $D=\{(x_1,y_1)\cdots(x_n,y_n)\}$
Expected imbalance ratio $IR^{'}$
Output: Resampled balanced trainset $D^{'}$
1. Calculate expected minority instance number, i.e. $N_{min}^{'} = (N_{maj} + N_{min}) / (1 + IR^{'})$
2. Calculate expected majority instance number, i.e. $N_{maj}^{'} = N_{maj} + N_{min} - N_{min}^{'}$
3. Synthesis $N_{min}^{'} - N_{min}$ minority instances using Algorithm 3
4. Remove $N_{maj}^{'} - N_{maj}$ majority instances using Algorithm 4.
5. Output all minority instances and remained majority instances as $D^{'}$

EXPERIMENTS AND DISCUSSION

All experiments in this paper are run on a PC with Intel (R) Core (TM) i5-3210M CPU @ 2.50G Hz, 10G memory, 500G hard disk capacity. We use Java as programming language with JDK version 1.7.0 and Eclipse as development tool. Basic classification algorithms are adopted from open source implementation of Weka 3.7 and we use Naive Bayes, Decision Tree and Support Vector Machine as basic algorithms in our experiments. In addition, we use 5-folds cross-validation method. Firstly, the dataset is randomly divided into roughly equal 5 subsets. Then, take each four subsets as train set and the remaining one as test set. Finally, average result of four time runs is taken as final result.

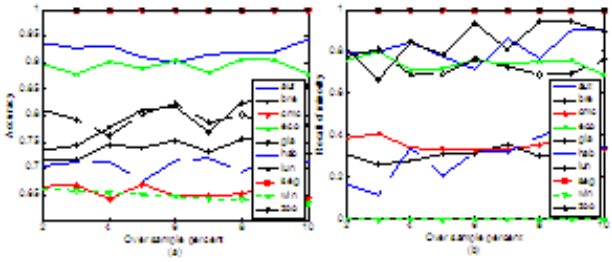

Fig. 1: Impact of over-sample ratio on (a) accuracy and (b) recall of minority

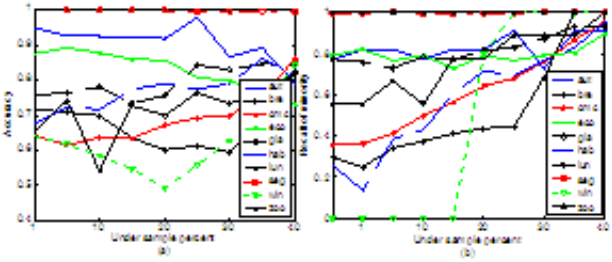

Fig. 2: Impact of under-sample ratio on (a) accuracy and (b) recall of majority

It is necessary to point out that datasets we used in data resample experiment and IMCC framework experiment is not the same. Mainly on account of: (1) use different datasets to reduce interference between each other method; (2) datasets used in resample experiment are of two-class, while datasets used in IMCC framework are of multi-class, these two kinds datasets are quite different.

Evaluation metric

For IMCC problem, traditional evaluation metrics are not fully applicable, such as accuracy, which can evaluate the overall classification result, but cannot distinguish contribution between minority and majority. In our experiments, we select 10 metrics to evaluate classification performance, including accuracy, precision of minority, recall of minority, F1 of minority, AUC of minority, precision of majority, recall of majority, F1 of majority, AUC of majority and G-mean which is the geometric mean of recall of minority and recall of majority.

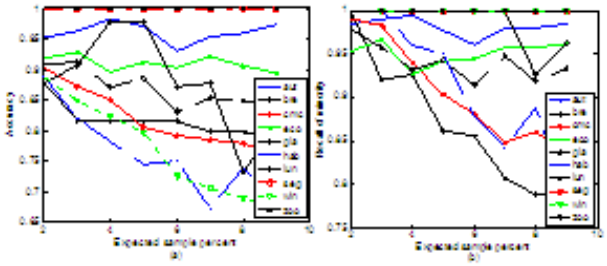

Fig. 3: Impact of expected imbalance ratio on (a) accuracy and (b) recall of minority

Self-adaptive data resampling

We select 10 datasets with different imbalance ratio from UCI machine learning repository. All those datasets are publicly available so that others can conduct same experiment. We especially select 3 medical datasets, namely breast-cancer, lung-cancer and haberman to test our method. In which, cmc is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey and are often used in pharmaceutical test. Remaining 6 datasets in other fields are selected to test the applicability of the algorithm. Detailed statistics of those datasets is shown in Table 1. For those datasets contain more than two categories, we first specify the minority category and then all other categories are merged together as majority. Finally we get 10 datasets with imbalance ratio range from 1.88 to 11.59.

Firstly, we test all parameters used in our algorithm to evaluate their impact on final results. Through experiment, we also find the optimal value of each parameter and it will be used in subsequent test. After that, we compare our data resample method with several other state-of-art methods.

Impact of over-sample ratio

The over-sample ratio is default set to 1 in Algorithm 3, which means over-sample percent is $(N_{maj} - N_{min}) / N_{min}$ and resampled dataset will contain roughly equal number of minority and majority instances. Here, we change the over-sample percent from 2 to 10 with step by 1 to see its impact on final result. We compared all those 10 evaluation metrics' value but only show accuracy and recall of minority for space limit. As it can be seen in Fig. 1(a), when the over-sample percent grow from 1 to 10, there is no significantly change of accuracy on most datasets. But accuracy on lung-cancer increase slowly, indicating that as more minority instances synthesized, minority instances in lung-cancer get more weight and its representation ability get enhanced, leading to improvement on overall accuracy. We can see from Fig. 1(b) that recall of minority increase gradually on most datasets which indicating that higher over-sample percent

can effectively enhance the weight of minority instances and improve recall of minority. Improvement on three medical datasets (breast-cancer, lung-cancer and haberman) is particularly evident, which is the very expected result. Results on other metrics show same trend and are not list here.

Impact of under-sample ratio

As show in Algorithm 4, under-sample ratio is set to 1 as default, which also aim at get a resampled dataset contains roughly equal number of minority and majority instances. In this case, the under-sample percent is $(N_{maj} - N_{min}) / N_{maj}$. When under-sample percent change from 1 to 40 with step by 5, accuracy and recall of minority on each dataset are show in Fig. 2. As it can be seen in Fig. 2(a), when over-sample percent increase, accuracy on segment always maintain as 1 which means all instances within segment are correctly classified. However, accuracy on haberman and ecoli decrease slowly all the time while accuracy on wine decrease at first and then increase. When compare statistics in Table 1, we can see that segment contains 2310 instances, in which there are 1980 majority instances. Under-sample on it does not significantly change its distribution, so accuracy on it maintains constant. However, for haberman and ecoli, their only contain 306 and 366 instances respectively. In addition, these two datasets only have 4 and 8 attributes respectively. When more and more majority instances are removed, misclassification of majority is significantly increased, resulting in decrease of accuracy. For most datasets, when under-sample percent beyond 20, accuracy increase gradually since more minority instances were correctly classified. When compare the recall of minority as show in Fig. 2(b), it can be seen that performance on all datasets except segment increase as under-sample percent increase. It shows that our method can effectively improve recall of minority which is of special importance in medical diagnostic.

Impact of expected sample ratio

In Algorithm 5, when expected sample ratio have settled, over-sample ratio and under-sample ratio will be calculated automatically, so that the final resampled dataset can achieve expected imbalance ratio and maintain the overall size of the resample dataset constant. Here, the key problem is the setting of expected sample ratio since in different applications, different user may focus on different category, and the expected sample ratio is not always the same. Here, we test the impact of expected sample ratio on each dataset and the results can be seen in Fig. 3. In out experiment, we set the minimum expected sample ratio as 1, i.e. resample dataset has roughly equal minority and majority instances. Since when number of majority instances reduces equal or less than number of minority instances, the original majority will becomes minority and the original minority becomes majority. So, we set expected sample percent from 1 to 9 with step by 1. As it can be seen from Fig. 3(a), when expected sample percent increase, accuracy on all datasets except autos shows a significant decline trend. So, we can conclude that if one aims at optimizing accuracy, it is better to set expected sample percent to 1. When compare the trend of recall of minority in Fig. 3(b), it shows the same trend which means as expected sample percent increase, less and less minority instances will be recalled. However, although not show here, recall of majority and precision of minority increase in some extent. So, if one aims at optimizing precision and recall of majority, higher expected sample ratio is a candidate choice.

Comparison with other methods

We compare our proposed methods, namely NBOS, DBUS and SIDR with baseline method that does not consider imbalance problem and classic imbalanced data resample method SMOTE (Chawla *et al*., 2002). Since there are 10 datasets and 10 evaluation metrics, we use Wilcoxon signed rank test (Rey and Neuhäuser, 2011) to determine whether our proposed methods can obtain significant better result.

As it can be seen from table 2, when the significance level α=0.05 ,our proposed SIDR method significant better than baseline method in precision of minority, recall of minority, F1 of minority, precision of majority, recall of majority and F1 of majority. It means that SIDR can cope with imbalance problem effectively. Compared with SMOTE, it shows that SIDR performs better than SMOTE in some metrics and comparable in other metrics, e.g. when the significance level α=0.05 , DBUS has significant better result than SMOTE on recall of minority. In addition, comparison among SIDR, NBOS and DBUS shows that combination method can obtain better result in most cases. It is necessary to point out that there is no perfect method in imbalanced classification filed and one should select most suitable method based on object to be optimized. As in our project, we need to provide advice for medical decision-making, recall of minority is our object to optimize. So, we incline to use our propose SIDR method.

IMCC Framework

In above section, we have tested our proposed data resample method and result shows that it is suitable to our application. In this subsection, we use SIDR as basic data resample method and test our proposed IMCC framework.

We select 13 datasets also from UCI machine learning repository and statistic of these datasets is show in Table 3. There are 7 datasets related to medical or pharmaceutical test, in which yeast and ecoli are used to predict the cellular localization sites of proteins, and lymphography, thyroid, dermatology, new-thyroid and contraceptive for medical diagnose.

Table 1: Statistic of datasets

Id	Name	Min/maj	# min/#maj	IR
cmc	cmc	Class 3/others	511/962	1.88
win	wine	Class 1/others	59/119	2.02
bre	breast-cancer	Class 2/class 1	85/201	2.36
lun	lung-cancer	Class 1/others	9/23	2.56
hab	haberman	Class 2/class 1	81/225	2.78
eco	ecoli	Class 2/others	77/259	3.36
zoo	zoo	Class 2/others	20/81	4.05
seg	segment	Class 2/others	330/1980	6
aut	autos	Class 3/others	22/183	8.32
gla	glass	Class 1/others	17/197	11.59

Table 2: Wilcoxon signed rank test

	Accuracy	Precision of minority	Recall of minority	F1 of minority	AUC of minority	Precision of majority	Recall of majority	F1 of majority	AUC of majority
SIDR vs Baseline	0.203125	0.011719	0.003906	0.003906	0.054688	0.011719	0.003906	0.003906	0.054688
SIDR vs SMOTE	0.496094	0.546875	0.250000	0.195313	0.820313	0.039063	0.203125	0.128906	0.820313
SIDR vs NBOS	0.652344	0.250000	0.039063	0.027344	1.000000	0.003906	0.019531	0.007813	1.000000
SIDR vs DBUS	0.359375	0.039063	0.019531	0.019531	0.300781	0.164063	0.019531	0.039063	0.300781
NBOS vs SMOTE	0.937500	0.843750	0.312500	0.382813	0.843750	0.546875	0.742188	0.945313	0.843750
DBUS vs SMOTE	0.148438	0.742188	0.027344	0.074219	0.128906	0.039063	0.496094	0.496094	0.128906

Table 3: Statistic of datasets

Id	Name	#Ins	#Class	IR
pag	page-blocks	5472	5	175.46
yea	yeast	1484	10	92.60
eco	ecoli	336	8	71.50
lym	lymphography	148	4	40.5
thy	thyroid	7200	3	40.16
aut	autos	159	6	16.00
gla	glass	214	6	8.44
bal	balance	625	3	5.88
der	dermatology	358	6	5.55
new	new-thyroid	215	3	5.00
hay	hayes-roth	160	3	2.10
con	contraceptive	1473	3	1.89
win	wine	178	3	1.48

Table 1: Wilcoxon signed rank test

IAOS vs	NB	DT	SMO
OVO	0.423828	0.002441	0.003418
OVA	0.423828	0.002441	0.003418
A&O	1.000000	0.034179	0.092285
ECOC	0.423828	0.002441	0.003418

It can be seen that dataset wine with the lowest imbalance ratio of 1.48 while dataset page-blocks with the highest imbalance ratio of 175.46. The number of instances each dataset contains also differs from each other, such as lymphography only contains 148 instances while thyroid contains 7200 instances. As for number of classes, it ranges from 3 to 10. In this stage, we carry out experiments on different datasets from different fields to test its applicability.

In this section, we compare our proposed IMCC framework (short as IAOS) with other state-of-art methods, include OVO, OVA, A&O and ECOC (Dietterich and Bakiri, 1995). Basic classification

algorithms we used include Naïve Bayes (NB), Decision Tree (DT) and Support Vector Machine (SVM). Sub-classifier output threshold is determined in advance through experiments and is set to λ=0.04 while imbalance ratio threshold is set to $0.4 \le T \le 0.6$.

We also carry out Wilkerson signed rank sum test between IAOS and other methods, the result is show in Table 4.

As can be seen from Table 4, when use Naive Bayes as basic algorithm, at the significance level α=0.05, there is no significant differences between IAOS and other methods. But when Decision Tree is used, at the significance level α=0.05, IAOS significantly better than other comparison methods, which means that IAOS performance significant better statistically. As for Support Vector Machine, at the significance level α=0.05, IAOS perform better than OVO, OVA and ECOC, but with no significant difference between A&O. When the significance level α=0.1, IAOS significant outperform all other methods. A&O can obtain similar results with IAOS because these two methods use similar data partition strategy. But, A&O does not take class imbalance into consideration and only select top two categories outputted by OVA while IAOS is designed especially for imbalanced dataset and select all categories that with output higher than threshold.

Overall, IAOS performs better in most cases, especially when there is more number of categories and higher imbalance ratio. Such as dataset yeast which contains 10 categories and the imbalance ratio is 92.60, accuracy obtained by IAOS can reach 57.88% while A&O can only reach 45.22%. IAOS improves the accuracy by 12.66%.

CONCLUSION

In medical diagnostic, pharmaceutical test and many other real world fields, there often exist IMCC problem. In such case, existing methods designed for balanced data may not get satisfactory results. In this paper, we propose an improved divide and conquer approach for IMCC problem, in which a framework for IMCC and a self-adaptive imbalanced data resample method are presented. Experiments on 23 UCI datasets, including 11 medical or pharmaceutical related datasets, show that our proposed method can effectively deal with IMCC problem and outperform compared methods, especially on medical and pharmaceutical datasets. In the paper, we default assuming that categories in dataset are parallel which means there are of equal weight. But in some actual applications, categories may in a hierarchical structure. In this case, imbalance between different level categories gets even serious and this will be the focus of our future work.

ACKNOWLEDGEMENTS

This work was supported by 973 Program (Grant No. 2013CB329601, 2013CB329602, 2013CB329604) and 863 Program (Grant No. 2012AA01A401, 2012AA01A402).

REFERENCES

Bunkhumpornpat C, Sinapiromsaran K and Lursinsap C (2011). DBSMOTE: Density-Based Synthetic Minority Over-sampling TEchnique. *APPL INTELL*, **36**: 664-684.

Bunkhumpornpat C, Sinapiromsaran K and Lursinsap C (2009). Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. *In*: *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, pp.475-482.

Chawla NV *et al.* (2002). SMOTE: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, **16**: 321-357.

Eitrich T *et al.* (2007). Classification of highly unbalanced CYP450 data of drugs using cost sensitive machine learning techniques. *J. Chem. Inf. Model.*, **47**: 92-103.

Fernández A *et al.* (2013). Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *KNOWL-BASED SYST*, **42**: 97-110.

Fernández A, Jesus MJ del and Herrera F (2010). Multi-class imbalanced data-sets with linguistic fuzzy rule based classification systems based on pairwise Learning. *In*: *Computational Intelligence for Knowledge-Based Systems Design*. Springer Berlin Heidelberg, pp.89-98.

Ghanem AS, Venkatesh S and West G (2008). Learning in imbalanced relational data. *In: 19th International Conference on Pattern Recognition*. IEEE, pp.1-4.

Ghanem AS, Venkatesh S and West G (2010). Multi-class Pattern Classification in Imbalanced Data. *In: 20th International Conference on Pattern Recognition*. IEEE, pp.2881-2884.

Han H, Wang WY and Mao BH (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. *In: Advances in Intelligent Computing*. Springer Berlin Heidelberg. pp.878-887.

He H *et al.* (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *In: IEEE International Joint Conference on Neural Networks*. IEEE World Congress on Computational Intelligence, pp.1322-1328.

Hoens TR *et al.* (2012). Building decision trees for the multi-class imbalance problem. *In: Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, pp.122-134.

Li X *et al.* (2014). In silico prediction of chemical acute oral toxicity using multi-classification methods. *J. Chem. Inf. Model.*, **54**: 1061-1069.

Luis M and Jesus AG (2006). Machine learning for imbalanced datasets: Application in medical diagnostic. *In: Flairs Conference.* pp.574-579.

Murphey YL *et al.* (2007). OAHO: An effective algorithm for multi-class learning from imbalanced Data. *In: International Joint Conference on Neural Networks*. IEEE, pp.406-411.

Phoungphol P *et al.* (2012). Multiclass SVM with ramp loss for imbalanced data classification. *In: 2012 IEEE International Conference on Granular Computing (GrC)*. IEEE, pp.376-381.

Prachuabsupakij W and Soonthornphisaj N (2012). Clustering and combined sampling approaches for multi-class imbalanced data classification. *In: Advances in information technology and industry applications*. Springer Berlin Heidelberg, pp.717-724.

Rifkin R and Klautau A (2004). In defense of one-vs-all classification. *J. Mach. Learn. Res.*, **5**: 101-141.

Sahin Y, Bulkan S and Duman E (2013). A cost-sensitive decision tree approach for fraud detection. *EXPERT SYST APPL*, **40**: 5916-5923.

Sanz JA *et al.* (2014). Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system. *APPL SOFT COMPUT*, **20**: 103-111.

Sun Y, Kamel MS and Wang Y (2006). Boosting for learning multiple classes with imbalanced class distribution. *In: Proceedings of the Sixth International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, pp.592-602.