Building a Memetic Algorithm Based Support Vector Machine for Imbalaced Classification

Wu Mingnan[1], Junzo Watada[2],,
Graduate School of Information, Production and Systems
Waseda University
[1] akikusu@gmail.com, [2] junzow@osb.att.ne.jp

Zuwarie Ibrahim[3], and Marzuki Khalid[4]
Centre of Artificial Intelligence and Robotics (CAIRO)
University of Technology Malaysia, Malaysia
[3] zuwairie@fke.utm.my, [4] marzuki@utm.my

Abstract— **Classification analysis is one of core research topics in pattern recognition field. According to the distribution of samples, algorithms like artificial network (ANN) and support vector machine (SVM) have been proposed to perform binary classification. But these traditional classification algorithms hardly work well for imbalanced dataset. In this study, a novel model on the basis of memetic algorithm (MA) and support vector machine (SVM) is proposed to perform the classification for large imbalanced dataset. It is named MSVC (memetic support vector classification) model. Memetic Algorithm is recently proposed and used as a heuristic framework for the large imbalanced classification. Because of the high performance of SVM in balanced binary classification, support vector classification (SVC) is combined with MA to improve the classification accuracy. G-mean is used to check the final result. Compared with some conventional models, the results showed that this model is a proper alternative for imbalanced dataset classification, and it expends the applications of memetic algorithm.**

Keywords- memetic algorithm (MA), support vector machine (SVM), classification on imbalanced dataset, memetic support vector classification (MSVC)

I. INTRODUCTION

In the real world, the dataset usually contains more instances from one class than another one. The class with more instances is referred as majority class, and the other one minority class. Applications like health control, vision recognition and credit card fraud detection focus their attention on the minority class. Most conventional algorithms try to minimize the error rate, ignoring the differences between majority class and minority class. So they cannot perform well on imbalanced dataset classification.

In this study, a novel algorithm is proposed for large imbalanced dataset classification based on memetic algorithm (MA) and support vector machine (SVM) and it is tested using some semiconductor data obtained from Intel. After pretreatment to the training data, memetic algorithm helps us get an optimized partition for the training space. As a heuristic framework, memetic algorithm also runs for selecting parameters of SVM classification. Although being different from genetic algorithm (GA) [1], memetic algorithm provides a proper alternative as an evolutionary algorithm (EA) [2]. This study extends the application of memetic algorithm, and the result shows the feasibility.

This paper is organized as follows. In Section I we introduce the research topics in this area briefly. In the Section II, we introduce some background knowledge about imbalanced dataset, and memetic algorithm (MA). After the introduction to basic knowledge, we describe the model proposed to deal with this classification in Section III. This model is named memetic support vector classification (MSVC) model. In Section IV, we classify the process to run this experiment under our proposal step by step. In Section V, results are pasted and we make some comparisons with some other existing proposals. It comes to the conclusion in the Section VI.

II. BACKGROUND KNOWLEDGE

2.1 Imbalanced dataset

2.1.1 Classification on imbalanced dataset

There are two kinds of imbalances used to be found in a binary classified dataset. One is named as between-class imbalance; another is named as within-class imbalance. In the case of between-class imbalance, one class has much more examples than the other class; in the case of within-class imbalance, some subsets of one class have much fewer examples than other subsets of the same class [3]. In imbalanced data sets, classes having more examples are called the majority classes and the ones having fewer examples are called the minority classes.

In many real world domains imbalanced datasets do exist. For examples, spotting unreliable telecommunication customers, detection of oil spills in satellite radar images and detection of fraudulent telephone calls and so on. In these real applications, the ratio of the minority class to the majority class can be drastic such as 1 to 1000, or even more. This factor is named as Imbalance Ratio (IR) while another factor is named as Lack Information (LI), which means that instances in the minority class are quite rare. Take the following two datasets as an example; one has 100:10000 instances, while another one has 10:1000 instances. Although the IR of both the dataset are the same, the LIs are different. Considering the LI factor, the previous dataset is easier to be accurately classified.

2.1.2 Evaluation of classification on imbalanced dataset

Let us classify the imbalanced dataset as class positive (P) and negative (N). Thus we can obtain Table 1,

TABLE 1 CONFUSION MATRIX

	PPos	PNeg
APos	TPos	FPos
ANeg	FNeg	TNeg

where PPos stands for Predicted Positive (PPos), PNeg stands for Predicted Negative (PNeg), APos stands for Actual

978-0-7695-4449-6/11 $26.00 © 2011 IEEE
DOI 10.1109/ICGEC.2011.95

Positive (APos), TPos stands for True Positive (TPos), FPos stands for False Positive (FPos), ANeg stands for Actual Negative (ANeg), FNeg stands for False Negative (FNeg), TNeg stands for True Negative (TNeg).

Majority Class is usually notated as Positive, and Minority Class is usually notated as Negative. Thus we can easily calculate the true positive rate (TPR) and true negative rate (TNR) by the following equations respectively:

$$TPR = \frac{TPos}{TPos + FNeg} \quad (1)$$

$$TNR = \frac{TNeg}{TNeg + FPos} \quad (2)$$

And another term, correctly classified rate (CCR) can be obtained by the following equation:

$$CCR = \frac{TPos + TNeg}{APos + ANeg} \quad (3)$$

CCR can assess the classification result well if APos is almost the same as ANeg. But as in imbalanced dataset, even if a high CCR is achieved, it might be contributed a lot by the majority class, with a high value of TPR, and low value in TNR. CCR doesn't meet the need for evaluating the classification on imbalanced dataset. Another way is to use geometric mean (G-mean), which is defined as the following equation:

$$G - mean = \sqrt{TPR \times TNR} \quad (4)$$

G-mean enables us to evaluate the classifier's performance fairly for an imbalanced dataset; at least it is much better than CCR.

2.2 Memetic Algorithm

Memetic Algorithm (MA) (Moscato, 1989) was firstly viewed as being close to a form of population-cased hybrid genetic algorithm coupled with an individual learning procedure capable of performing local refinements. It was not very far after the theory of Universal Darwinism (Richard Dawkins, 1983), which suggests that evolution is not limited to biological system, not confined to genes, but also applicable to any complex system that exhibits the principles of inheritance, variation and selection (Wikipedia). The term "meme" (Richard Dawkins, 1976) is defined as "the basic unit of cultural transmission, or imitation", and also as "an element of culture that may be considered to be passed on by non-genetic means".

Pseudo code of MA:

- *Process of memetic algorithm:*
- *Define: stopping condition s, number of individuals n;*
- *Initialize: Generate an initial population P;*
- *While (s is false) do*
- *{*
    - *Evolve new individuals by stochastic offset;*
    - *Evaluate every individual i in the total population;*
    - *Select the best n individuals W;*
    - *for w in W*
    - *{*
    - *Perform individual learning using meme;*
    - *Proceed with individual learning;*
    - *}end for;*
- *}end while;*

In the section of individual learning, meme is defined according to the problem domain. Meme can be defined as a real value, a binary value, a vector and so on. Later in the MSVC model we will explain the meme in this imbalanced dataset classification. In the section of "proceed with individual learning", usually Lamarckian (Lamarck, 1794-1796) learning or Baldwinian (Baldwin, 1896) learning is used. A Lamarckian example is that "Giraffes stretching their necks to reach leaves high in trees, strengthen and gradually lengthen their necks. These giraffes have offspring with slightly longer necks". While a Baldwinian example is that "Individuals who learn the behavior to escape from predator more quickly will obviously be at an advantage. As time goes on, the ability to learn the behavior will improve (by genetic selection), and at some point it will seem to be an instinct". We will explain the learning process later in the MSVC model according to this certain problem.

MA is different from genetic algorithm (GA) (Goldberg, 1994). The biggest difference lies in that MA uses meme as the heuristic unit, while GA uses gene as the heuristic unit. In computation, genes are usually denoted by binary value or real value. Memes can be expressed by an object. MA can perform well no matter what value are assigned to memes. Another difference is that MA and GA use different ways to evolve new individuals. GA uses crossover and mutation to generate new individuals, and this process is like a stochastic process. MA generates new individuals after learning, and it can be said this process is under some rules. Theoretically, MA can optimize more efficiently than GA, which means in less generations, generating better results.

III. PROBLEM DOMAIN

We got some encrypted semiconductor dataset from Intel, and we perform out experiments based on this dataset. According to the introduction from Intel, the minority implies the defective products and the majority implies the qualified products. If we can find out these defective products, we can guarantee the rate of qualified products. We should never classify the minority instance as majority instance.

An instance of this dataset consists of four attributes, with two numerical input parameters, one categorical parameter and one output.

Two typical instances are shown in Table 2.

TABLE 2 TWO REAL INSTANCES

output	categ	num1	num2
0	103A	236780	236400
1	134A	262540	260630

Feature num1 ranges from 174610 to 299720; feature num2 ranges from 174650 to 301800. The categorical parameter categ consists of 210 different types ranging from 1A to 210A. The output is the classification information, notated as 1 and 0. Please check it in Table 3.

TABLE 3 SEMICONDUCTOR DATA DETAIL

attribute	num1	num2	categ	output
Training data	174610-261970	174650-278140	1A – 210A	0,1
Testing data	180800-299370	176670–301770	1A – 210A	0,1

The data in training dataset and testing dataset are imbalanced. The amount of samples with label 0 is much greater than samples with label 1 in each dataset. The data with label 0 are the majority class, and the data with label 1 are the minority class. Tables 4 and 5 show some features of the dataset. The number is the account of instances in each class.

TABLE 4 TRAINING DATA STATUS

Majority Class	MinorityClass	Total	Minority/Total
107424	8890	116314	0.076

TABLE 5 TESTING DATA STATUS

Majority Class	Minority Class	Total	Minority/Total
107529	8821	116350	0.076

IV. MSVC MODEL

In this proposed MSVC model, MA provides a heuristic framework to find an optimal partition of the original training space and SVM is used to perform classification in a certain partition. In the process of SVM, another MA optimization is performed to find the optimal parameters C and γ for SVM.

The following is the pseudo code for memetic algorithm in the heuristic framework.

- *Step 1: scale the data in training dataset*
  num1=num1/max (numerical 1)*
  num2=num2/max (numerical 2)*
- *Step2: code the training space into distribution blocks*
- *Step3: define the stopping condition s, number of partitions p and number of memes n.*
- *Step4: generate n initial p partition memes randomly, and train them using SVM*
- *Step5: evolve n new memes based on n previous memes by stochastic/ruled offset, and number every one, training them using SVM;*
- *Step6: test these memes using testing dataset; calculate the fitness of all these 2n memes after SVM's classification*
- *Step7: select n best memes W*
- *Step8: for every w in W, classify the differences between every pair of previous meme and generated meme, which is a learning process.*
- *Step9: update the rule base according to the learning result.*
- *Step10: if the s is satisfied, terminate the computation; if not, go back to step 4.*

The above procedure is the process of MSVC. After generations of imitations, or some computing threshold, it comes to an end.

Scaling is to make the data usable for SVM classification.

Here meme consists of memeID and partition information. About the partition information, we will explain clearly in the experiment section.

In this total procedure, the parameter stopping condition s, numbers of partitions p, number of memes n are crucial to control the whole computation. These three parameters determine the final computation time and classification accuracy.

In step5, in the first circle, new memes are evolved by a randomly way. After the first circle, since we can obtain some rules in step8 and step9, we will evolve new memes under these rules. In this step, offset is something like changing the combinations of every partition. In the experiment section, we will explain it more clearly in this certain situation. For different situations, offsets can be totally different. For example, in the situation of finding an optimal result for a numerical computation, offset can be simply a random real value between 0 and 1. While in the situation of a travelling salesman problem (TSP) [6], offset can be a binary value. Application domains verify the offsets.

Each pair of memes has a parent meme and a child meme. Comparisons in step8 are performed in every pair to confirm what makes higher fitness. These experiences form some rules, and these rules are stored in rule base, in the form of list.

Since some memes have already gotten their fitness in the previous computation circle, these memes will be marked. Thus in step6, we can reduce the computation load by this way.

In the SVM classification section, an MA optimization is performed to find the optimal parameters C and γ for SVM. The following is the procedure of this MA optimization.

- *Step1: define the stopping condition s, number of meme n.*
- *Step2: initialize n (C,γ) memes randomly*
- *Step3: evolve n new memes based on n previous memes by stochastic/ruled offset, and number every one;*
- *Step4: calculate the fitness of all these 2n memes*
- *Step5: select n best individuals as memes W*
- *Step6: for every w in W, classify the differences between every pair of previous meme and generated meme, which is a learning process.*
- *Step7: update the rule base according to the learning result.*
- *Step8: if the s is satisfied, terminate the computation; if not, go back to step 3.*

The above procedure is the MA process of SVM. After generations of imitations, or some computing threshold, it comes to an end.

Here meme consists of memeID and two real value C and γ. Offset is just some random values; these random values change the C and γ after a treatment.

V. EXPERIMENT AND RESULT

Before we perform the classification, we do some pretreatment on the dataset. As shown in the Figure 2, we grid the training space into different units. This demo grids the

original space into a 7 times 6 matrix. Some blocks have no distribution, so they don't count. In these 42 blocks, there are totally 15 blocks that are with of distribution, we number them for partition.

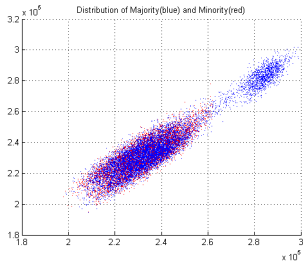

FIGURE 2 DISTRIBUTION BLOCK

The MA process in SVM is very simple since it just concerns some numerical computation. The stopping condition is different with the condition in MSVC, where here if one computation takes 8 seconds, it comes to an end. Offset here is simply real value number between -1 and 1, notated as *r*. New parameter is evolved as *C*(1+r)* and it is the same as γ.

In Table 6, we listed out some common notations used in experiment.

TABLE 6 NOTATIONS FOR COMPUTATION

Attribute	Meaning
num1*	num1/max (numerical 1)
num2*	num2/max (numerical 2)
s	Stopping conditions (50 circles computation)/ 8 seconds computation
p	Number of partition
n	Number of memes
C	Parameter C for SVM
γ	Parameter γ for SVM
offset	Generated randomly /under a certain rule
W	Candidate memes

We totally performed five experiments on this dataset as shown in Table 7. In the first experiment, the training space was grid into 32 by 32 matrix, with 8 partitions, 8 memes. In the second experiment, the training space was grid into 32 by 32 matrix, with 16 partitions, 8 memes. In the third experiment, the training space was grid into 64 by 64 matrix, with 8 partitions, 8 memes. In the forth experiment, the training space was grid into 32 by 32 matrix, with 8 partitions, 16 memes. In the fifth experiment, the training space was grid into 256 by 256 matrix, with 32 partitions and 32 memes. Stopping condition is that they should finish 50 circles calculation. The MA optimization for every SVM's parameters was uniformly set as 8 seconds computation.

TABLE 7 RESULTS

Experiment	TPR	TNR	G-mean
First	81.3%	84.7%	83.0%
Second	87.4%	85.1%	86.2%
Third	88.3%	86.2%	87.2%
Forth	81.9%	85.1%	83.4%
Fifth	89.2%	88.9%	89.0%

Through these experiments, we find that the MSVC model is able to deal with the imbalanced classification problem. All these TPR and TNR are higher than 0.8, neither of TPR nor TNR is neglected.

To evaluate the performance of MSVC, we compare the best experiment result with other model in [4][5] and some other research results by Watada, et al as shown in Table 8.

TABLE 8 COMPARISON

Model	G-Mean	Model	G-mean
SVM-RBF	79.0%	BN	82.2%
RSA-SVM	90.1%	GBT	88.9%
MIPS	87.8%	TSLM-ANN	87.9%
MSVC	89.0%	ANN	63.4%

Although the MSVC didn't provide the best result in the listed models, the comparisons showed that it is a proper alternative for classification since the G-mean accuracy is high and MSCV deals with both majority and minority fairly. But MSVC can execute the computation procedure fewer steps than RSA-SVM. Besides, this implementation extends the application of memetic algorithm (MA). It is meaningful. Still, Memetic Algorithm should be developed in other fields.

VI. CONCLUSIONS

Data mining on Imbalanced dataset is important, including the classification problem. In this paper, we reviewed the imbalanced dataset classification, memetic algorithm and support vector machine, and then proposed a novel model named memetic algorithm based support vector machine for classification (MSVC model). G-mean was used to test the applicability of proposed MSVC model.

The data used here is semiconductor data from Intel. It was used for both training and testing. After five experiments, it was proved that this MSVC model provides a proper alternative for imbalanced classification problem. MSVC model can classify majority and minority fairly. And also, this study extended the applications of memetic algorithm, which is most meaningful.

REFERENCES

593–604

[1] Goldberg, David E. (1989). Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley. 41.

[2] Eiben, A.E., Smith, J.E. (2003), Introduction to Evolutionary Computing, Springer.

[3] Qiong Gu,Zhihua Cai,Li Zhu, Bo Huang (2008) Data Mining on Imbalanced Data Sets, *International Conference on Advanced Computer Theory and Engineering*

[4] Lei Ding, Junzo Watada, Lim Chun Chew, Zuwairie Ibrahim, Lee Wen Jau and Marzuki Khalid (Dec. 2010), "A SVM-RBF Method for Solving Imbalanced Data Problem," ICIC EL, 4 (6(B)) 2419-2424

[5] Junzo Watada, Lee-Chuan Lin, Lei Ding, Mohd Ibrahim Shapiai, Lim Chun Chew, Zuwairie Ibrahim Lee Wen Jau, and Marzuki Khalid (2010), "A Rough-Set-Based Two-class Classifier for Large Imbalanced Dataset," Smart Innovation, Systems and Technologies, 1, Volume 4, Advances in Intelligent Decision Technologies, XIII., 641-651

[6] Applegate, D. L.; Bixby, R. M.; Chvátal, V.; Cook, W. J. (2006), *The Traveling Salesman Problem: A Computational Study*, Princeton University Press, ISBN 0691129932.