Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
DAAAM Symposium

IMPROVING CLASSIFICATION WITH COST-SENSITIVE METACLASSIFIER

MUNTEAN, M[aria]; VALEAN, H[onoriu]; ILEANA, I[oan]; RISTEIU, M[ircea] & JOLDES, I[ulian]

Abstract: *This paper introduces the Enhancer, a new algorithm that improves the Cost-sensitive classification for Support Vector Machines, by multiplying in the training step the instances of the underrepresented classes. We have discovered that by oversampling the instances of the class of interest, we are helping the Support Vector Machine algorithm to overcome the soft margin. As an effect, it classifies better future instances of this class of interest.*

Key words: *classification; accuracy; metaclassifier, algorithm*

1. INTRODUCTION

Most of the real-world data are unbalanced in terms of proportion of samples available for each class, which can cause problems such as over fit or little relevance. The Support Vector Machine (SVM), proposed by Vapnik and his colleagues in 1990's [Vapnik, 2000], is a new machine learning method based on Statistical Learning Theory and it is widely used in the area of regressive, pattern recognition and probability density estimation due to its simple structure and excellent learning performance. Joachims validated its outstanding performance in the area of text categorization in 1998. SVM can also overcome the over fitting and under fitting problems [Hong et al., 2009], and it has been used for imbalanced data classification [Li et al., 2009].

A SVM is an algorithm that uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyper plane. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyper plane. The SVM finds this hyper plane using support vectors and margins.

2. COST-SENSITIVE CLASSIFICATION

In actual applications, it exist the problems that wrong classify result in different harm degree of different sort sample. The solution proposed in literature is the Cost-sensitive SVM approach [He et al., 2009], a new method for unbalanced classification.

Fundamental to the Cost-sensitive learning methodology is the concept of the cost matrix. This approach takes the classify cost into account, and it aims to reduce the classify cost to the least. Instead of creating balanced data distributions through different sampling strategies, Cost-sensitive learning targets the imbalanced learning problem by using different cost matrices that describe the costs for misclassifying any particular dataset. A very useful tool, the Confusion Matrix for two classes is shown in Table 1.

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	TP	FN
	Class = 0	FP	TN

Tab. 1. Confusion matrix for a two-class problem

The true positives (TP) and true negatives (TN) are correct classifications. A false positive (FP) occurs when the outcome is incorrectly predicted as 1 (or positive) when it is actually 0 (negative). A false negative (FN) occurs when the outcome is incorrectly predicted as negative when it is actually positive.

In addition, the accuracy measure may be defined. It represents the ratio between correctly classified instances and the sum of all instances classified, both correct and incorrect ones. The above measure was defined as:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

3. DESCRIPTION OF THE MULTIPLYING ALGORITHM USED

This paper introduces an algorithm named Enhancer aimed for increasing the TP of underrepresented classes of datasets, using Cost-sensitive classification and SVM.

Experimentally we have found out that the features that help in raising the TP of a class are the cost matrix and the amount of instances that the class has. The last one can be modified by multiplying the number of instances of that class that the dataset initially has.

The Enhancer algorithm is detailed in the following pseudo code:

```
1. Read and validate input;
2. For all the classes that are not well represented:
   BEGIN
    Evaluate class with no attribute added
    Evaluate class at Max multiplication rate
    Evaluate the class at Half multiplication
    REPEAT
        Flag  = False
        Evaluate the intervals (beginning, middle),
        (middle, end)
        If the end condition is met
        Flag = True
         If the first interval has better results we should use
         this, otherwise the other
        Find the class evaluation after multiplying class
        instances middle times
    UNTIL Flag = False
    END
3. Multiply all the classes with the best factor obtained;
4. Evaluate dataset.
```

The Enhancer algorithm described in the pseudo code used a *Divide et Impera* technique, that searched in the space (0 multiplication – max multiplication) for the optimal multiplier for the class. The algorithm is going to stop its search under two circumstances:

- The granulation is getting to thin, i.e., the difference between the beginning and end of an interval is very small (under a set epsilon). This constraint is set, in order not to let the algorithm wonder around searching for solutions that vary one from another by a very small number ($<10^{-2}$).

- The modulus of the difference between the $\Delta TP_i + \Delta A_{CC}$ from the first and the second interval should be bigger that a known value. This value is the considered to be the deviation of the Accuracy added to the deviation of the TP of that class:

$$\mu = \sigma A_{CC} + \sigma TP \tag{2}$$

With the 10 fold cross validation, the dataset was randomized, and stratified using an integer seed that took values in the range 1-10. The algorithm performed 10 times the evaluation of the data set, and all the time had a different test set.

4. EXPERIMENTAL RESULTS

For the evaluation, we used the Pima dataset, obtained from the online UCI Machine Learning Repository. The class distribution for this dataset is illustrated below (Fig. 2):

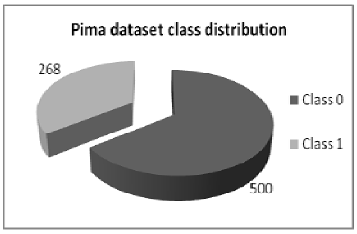

Fig. 2. Pima dataset class distribution

In order to improve the classification of the weakly represented classes in those datasets, in which they are in very small numbers with respect to the other classes, two approaches were tested:

- Cost-sensitive classification
- Multiplication of the instances of weakly represented classes

A. *Cost-sensitive classification*

By modifying the cost matrix, we obtained a variation quite high in the TP of Class 0 and we were able to change the level of the accuracy from 41% to 89% (Fig. 3).

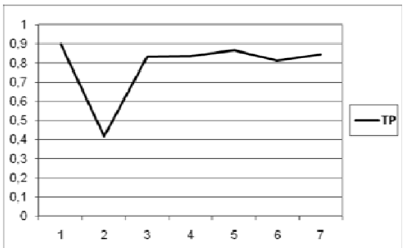

Fig. 3. Class 0 TP variation with respect to Cost Matrix change

The TP of the Class 0 and Class1 evolve almost complementary one from another (when one TP rises, the other falls and the other way around). The TP of class 1 also seemed to be bounded by about 96%.

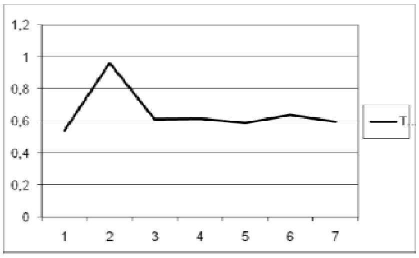

Fig. 4. Class 1 TP variation with respect to Cost Matrix change

B. *Multiplying underrepresented classes*

After applying the class multiplications all the TP of Class 1 hits a zone of instability, until the multiplying factor reached 1.0, when the TP ascent stabilized. Seemingly the problem in this case is that the accuracy was dropping or remaining constant, while the TP of the Class reached the maximum value (Fig. 5).

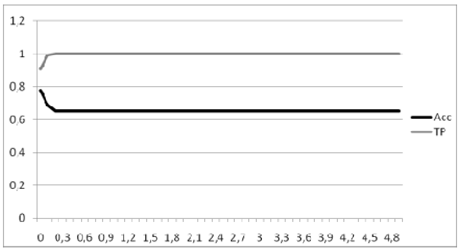

Fig. 5. The evolution of the TP of Class 1 and the general accuracy with respect to the no. of instances of Class 1

5. CONCLUSIONS

We observed that the Enhancer classifier performed better than Cost Sensitive classifier (Fig. 6) and that with the new algorithm, the TP of certain classes of interest were increased significantly while keeping the general accuracy in the desired range.

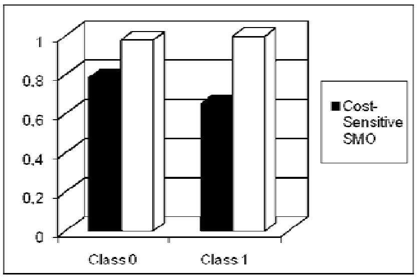

Fig. 6. Comparison between the TP of the classes resulting Cost Sensitive SMO Evaluation and with the Enhancer

We have also discovered that by oversampling the instances of the class of interest, we are helping the SVM algorithm to overcome the soft margin. As an effect, it classifies better future instances of this class of interest.

This solution is especially important when it is far more important to classify the instances of a class correctly, and if in this process we might classify some of the other instances as belonging to this class we do not produce any harm.

6. REFERENCES

He, H. & Garcia, E., A., Learning from imbalanced data (2009). *IEEE Transactions on Knowledge and Data Engineering*, VOL. 21, NO. 9, September, 2009, ISSN: 1041-4347

Hong, M.; Yanchun, G.; Yujie, W. & Xiaoying, L. (2009). Study on classification method based on Support Vector Machine, *2009 First International Workshop on Education Technology and Computer Science*, pp.369-373, March, 7-8, 2009, ISBN: 9781424435814, Wuhan, China

Li, Y.; Danrui, X. & Zhe, D. (2009). A new method of Support Vector Machine for class imbalance problem, *2009 International Joint Conference on Computational Sciences and Optimization*, pp. 904-907, April 24-26, 2009, ISBN: 9780769536057, Hainan Island, China

Vapnik, V., N. (2000). The nature of statistical learning theory, NewYork: *Springer-Verlag*, 2000, ISBN: 9780387987804

*** (2010) http://archive.ics.uci.edu/ml/ - University of California Irvine. UCI Machine Learning Repository, *Accessed on: 2010-06-15*