Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2577

Applying One-sided Selection to Unbalanced Datasets

Gustavo E.A.P.A. Batista, André C.P.L.F. Carvalho, Maria Carolina Monard.

Nº 56

NOTAS DO ICMC
Série Computação

São Carlos
Dez./2000

Applying One-sided Selection to Unbalanced Datasets

Gustavo E.A.P.A. Batista[1], André C.P.L.F. Carvalho[1], and Maria Carolina Monard[2]

[1] Departamento de Computação
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - Campus de São Carlos
Caixa Postal 668
13560-970 São Carlos, SP
[2] Instituto de Ciências Matemáticas e de Computação/ILTC
Universidade de São Paulo - Campus de São Carlos
Caixa Postal 668
13560-970 São Carlos, SP
Phone: +55-16-273-9692. FAX: +55-16-273-9751.
{gbatista, andre, mcmonard}@icmc.sc.usp.br

Abstract. Several aspects may influence the performance achieved by a classifier created by a Machine Learning system. One of these aspects is related to the difference between the number of examples belonging to each class. When the difference is large, the learning system may have difficulties to learn the concept related to the minority class. In this work[1], we discuss some methods to decrease the number of examples belonging to the majority class, in order to improve the performance of the minority class. We also propose the use of the VDM metric in order to improve the performance of the classification techniques. Experimental application in a real world dataset confirms the efficiency of the proposed methods.

1 Introduction

Supervised learning is the process of automatically creating a classification model from a set of instances, called the *training set*, which belong to a set of classes. Once a model is created, it can be used to automatically predict the class of other unclassified instance.

In other words, in supervised learning, a set of n training examples is given to an inducer. Each example $\mathbf{X}$ is an element of the set $F_1 \times F_2 \times \ldots \times F_m$ where F_j is the domain of the jth feature. Training examples are tuples $(\mathbf{X}, Y)$ where Y is the label, output or class. The Y values are typically drawn from a discrete set of classes $\{1, \ldots, K\}$ in the case of *classification* or from the real values in the case of *regression*. Given a set of training examples, the learning algorithm (*inducer*)

[1] This work was previously published in the MICAI-2000, Lecture Notes in Artificial Intelligence, Vol. 1793, Springer-Verlag. Best Paper Award Winner.

outputs a *classifier* such that, given a new instance, it accurately predicts the label Y.

In this work we refer to classification and, for simplicity, we consider two class problems. However, the methods here described can easily be adapted and applied to problems with more than two classes.

One of the problems in supervised learning is learning from unbalanced training sets. For a number of application domains, for instance the diagnostic of rare diseases, a huge disproportion in the number of cases belonging to each class is common. In these situations, the design of a high precision classifier is straightforward: just classifying every new case as belonging to the majority class. However, accurate classification of minority class cases is frequently the major objective of such applications.

Many traditional learning systems are not prepared to induce a classifier that accurately classify both classes under such situation. Frequently the classifier has a good classification accuracy for the majority class, but its accuracy for the minority class is unacceptable.

In this work we discuss several methods for selecting training cases labelled with the majority class in order to improve the classification accuracy of the minority class. The main idea is to select and remove cases from the majority class while leaving untouched cases from the minority class. However, only cases that have low influence on learning the majority class concept are removed, such that classification accuracy of this class is not deeply affected. Removing cases from the majority class tends to decrease the disproportion between the classes, and as a result, to improve the minority class classification. Selection methods that focus on decreasing the number of cases from the majority class are known as *one-sided selection methods* [9].

In this work we propose some improvements to the one-sided selection methods proposed in [9] through the use of a powerful method to measure distance of symbolic attributes called *Value Difference Metric (VDM)*. Kubat's method is based on instance-based learning systems. This sort of learning systems typically store the training set and use a distance metric to classify new cases. An Euclidean distance can be used to compare examples with continuous attributes values. However, whenever the dataset has symbolic attributes, instance-based systems frequently use very simple metrics, such as overlap metric (same attribute values have distance equal to zero while different attribute values have distance equal to one). Metrics like overlap metric may fail to capture the complexity of the domain, causing an inadequate performance of one-sided selection methods.

This work is organised as follows: Section 2 gives a general idea of why several learning systems are not prepared to cope with an unbalanced dataset. Section 3 explains why error rate and accuracy are not good metrics to measure the performance of learning systems trained on unbalanced datasets, and describes other metrics to substitute them. Section 4 describes the one-sided selection methods and Section 5 briefly describes the VDM metric for symbolic attributes. Some initial experiments using the MineSet tool from SGI are described in section 6,

and performance measures are show in Section 7 in order to verify the effectiveness of the proposed methods. Finally, Section 8 shows the conclusions of this work.

2 Why Unbalanced Datasets Harm?

Learning from unbalanced datasets is a difficult task since most learning systems are not prepared to cope with a large difference between the number of cases belonging to each class. However, real world problems with these characteristics are common. Researchers have reported difficulties to learn from unbalanced datasets in several domains, such as retrieving information from texts [11], diagnosing rare diseases [4], detecting credit card fraud operations [14], and others.

Why learn under such conditions is so difficult? Imagine the situation illustrated in Figure 1, where there is a large unbalance between the majority class (-) and the minority class (+). It also shows that there are some cases belonging to the majority class incorrectly labelled (noise). Spare cases from the minority class may confuse a classifier like *k-Nearest Neighbour (k-NN)*. For instance, 1-NN may incorrectly classify many cases from the minority class (+) because the nearest neighbour of these cases are noisy cases belonging to the majority class. In a situation where the unbalance is very high, the probability of the nearest neighbour of a minority class case (+) be a case of the majority class (-) is near 1, and the minority class error rate will tend to 100%, which is unacceptable for many applications.

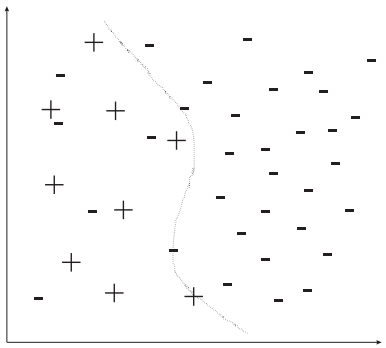

Fig. 1. Many negative cases against some spare positive cases.

Decision trees also suffer from a similar problem. In the presence of noise, decision trees may become too specialised (overfitting), i.e., the decision tree inducer may need to create many tests to distinguish the minority class cases (+) from noisy majority class cases. Pruning the decision tree does not necessarily alleviate the problem. This is due to the fact that pruning remove some branches considered too specialised, labelling new leaf nodes with the dominant class on

that node. Thus, there is a high probability that the majority class will also be the dominant class of those leaf nodes.

Theoretical analysis show that Multi-Layer Perceptron networks (MLP) approximate a posterior bayesian probability which is independent of prior bayesian probability. However, empirical analysis have shown that MLP neural networks have difficulty to learn when trained with unbalanced datasets [1]. According to [10], such disagreement between empirical and theoretical analysis is due to some suppositions made in theoretical analysis, such as a infinite training dataset, reaching the global minimum while training, and others.

3 Metrics to Measure Performance with Unbalanced Datasets

The error rate (E) and the accuracy (1-E) are widely used metrics for measuring the performance of learning systems. However, when the prior probability of the classes is very different, such metrics might be misleading. For instance, it is straightforward to create a classifier having 90% accuracy if the dataset has a majority class with 90% of the total number of cases, by simply labelling every new case as belonging to the majority class. Other factor against the use of accuracy (or error rate) is that these metrics consider different classification errors as equally important. For instance, a sick patience diagnosed as healthy might be a fatal error while a healthy patience diagnosed as sick is considered a much less serious error since this mistake can be corrected in future exams. On domains where misclassification cost is relevant, a cost matrix could be used. A cost matrix defines the misclassification cost, i.e. a penalty for making a mistake for each different type of error. In this case, the goal of the classifier is to minimise classification cost instead of error rate.

Different types of errors and hits performed by a classifier can be summarised as a *confusion matrix*. Table 1 illustrates a confusion matrix for two classes.

	Positive Prediction	*Negative Prediction*
Positive Class	True Positive (a)	False Negative (b)
Negative Class	False Positive (c)	True Negative (d)

Table 1. Different types of errors and hits for a two classes problem.

From such matrix it is possible to extract a number of metrics to measure the performance of learning systems, such as error rate $\frac{(c+b)}{(a+b+c+d)}$ and accuracy $\frac{(a+d)}{(a+b+c+d)}$. Other two metrics directly measure the classification error rate on the positive and negative class:

- False negative rate: $\frac{b}{a+b}$ is the percentage of positive cases misclassified as belonging to the negative class;

– False positive rate: $\frac{c}{c+d}$ is the percentage of negative cases misclassified as belonging to the positive class.

Figure 2 shows a common relationship among error rate, false positive rate and false negative rate. This analysis aimed to identify fraudulent credit card transactions (minority class, represented as negative class) into valid transactions (majority class, represented as positive class). Chan and Stolfo trained the C4.5 learning system [12] with different class distributions in the training class (the test class distribution was kept untouched) [5]. The chart begins with a training set consisting of 90% of cases from the majority class. The proportion of minority class cases is increased by 10% at each iteration. This increase in the number of minority class cases in the training set causes an improvement in the classification of cases from this class. However, the classification accuracy for the majority class decreased. The error rate on the test set, on the other hand, increased influenced by the bad performance of the majority class, which dominates the test set.

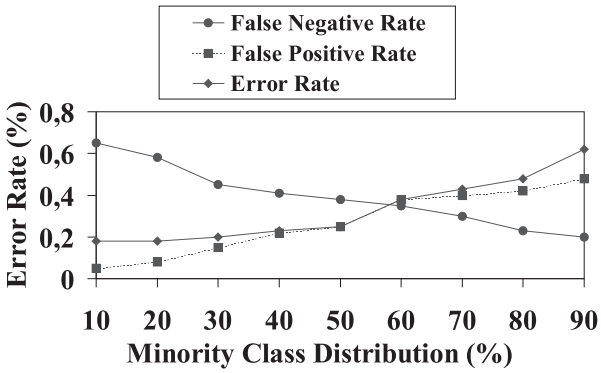

Fig. 2. Error rates with different class distributions in the training class.

This experiment showed a decrease in the majority class accuracy as more cases from the minority class were added to the training set. In real situations, when it is difficult to find more cases of the minority class, a more suitable solution is to remove majority class cases, in order to get a more balanced distribution among the classes. However, a question stands: is it possible to reduce the number of the majority class cases without loosing too much performance on the majority class? This is the main objective of one-sided selection.

4 One-sided Selection

The problem of unbalanced datasets has been studied and some solutions have been proposed. One of them, one-sided selection [9] proposes a careful removal

of cases belonging to the majority class while leaving untouched all cases from the minority class, because such cases are too rare to be lost, even though some of them might be noise. Such careful removal consists of detecting and removing cases considered less reliable, using some heuristics. These heuristics can be better understood by dividing the cases into four distinct groups:

1. Mislabelled cases (noise). For instance, the majority class cases (-) located in the left region of Figure 1;
2. Redundant cases. Such cases might be represented by other cases that are already in the training set. For instance, the cases that are far from the decision border, like those located on the right top region of Figure 1;
3. Cases close to the decision border (borderlines). These cases are quite unreliable since even a small quantity of noise can move them to the wrong side of the decision border;
4. Safe cases. Those cases that are neither too close to the decision border nor are too far from it. These cases should be kept for learning.

The one-sided selection technique aims to create a training set consisting of safe cases. In order to achieve that, noisy, borderline and redundant majority class cases should be eliminated.

Borderline and noisy cases can be detected by *Tomek links* [15]. Given the cases x and y belonging to different classes and be $d(x, y)$ the distance between x and y. A (x, y) pair is called a Tomek link if there is not a case z, such that $d(x, z) < d(x, y)$ or $d(y, z) < d(y, x)$. Cases that are Tomek links are borderline or noise. Figure 3 illustrates a dataset obtained by the removal of the majority class cases (see Figure 1) that formed Tomek links.

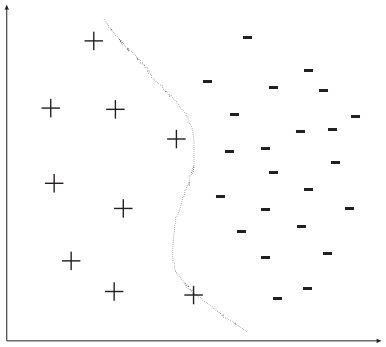

Fig. 3. Dataset without noisy and borderline cases.

Redundant cases can be deleted by finding a *consistent subset*. By definition, a consistent subset $C \subset S$ is consistent with S if, using a 1-nearest neighbour (1-NN), C correctly classify the cases in S [7]. An algorithm to create a subset C from S is the following: First, randomly draw one majority class case and all

cases from the minority class and put these cases in C. Afterwards, use a 1-NN over the cases in C to classify the cases in S. Every misclassified case from S is moved to C. It is important to note that this procedure does not find the smallest consistent subset from S. Figure 4 shows the dataset (see Figure 1) after the removal of redundant cases from the majority class.

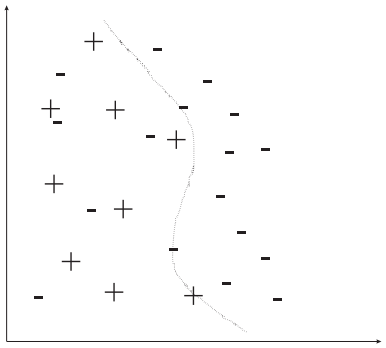

Fig. 4. Dataset without redundant cases.

5 The VDM Metric

In [13] a powerful method to measure distance of symbolic attributes called *Value Difference Metric (VDM)* is presented. In contrast to simpler methods, for instance the overlap metric, which measure the distance between symbolic attributes by just verifying if the attribute has the same value (distance equals to 0) or not (distance equals to 1). The VDM metric considers the classification similarity for each possible value of an attribute to calculate the distances between these values. As a result, a matrix of distances is created from the training set for each attribute. The distance d between two values for a certain symbolic attribute V is defined by:

$$d(V_1, V_2) = \sum_{i=1}^{n} \mid \frac{C_{1_i}}{C_1} - \frac{C_{2_i}}{C_2} \mid^k$$

In this equation, V_1 and V_2 are two values assumed by the attribute V. n is the number of classes in the dataset and C_{1_i} is the number of cases belonging to the class i whose attribute V assumed the value V_1. C_1 is the total number of cases whose attribute V assumed the value V_1 and k is a constant, frequently 1.

The VDM metric considers two values to be similar if they occur with almost identical relative frequencies for all classes. The ratio $\frac{C_{1_i}}{C_1}$ represents the probability of a case to be classified as belonging to the class i given that the attribute V has the value V_1.

The VDM metric, used in this work, has the following characteristics:

1. $d(a,b) > 0, a \neq b$;
2. $d(a,b) = d(b,a)$;
3. $d(a,a) = 0$;
4. $d(a,b) + d(b,c) \geq d(a,c)$.

6 Initial Experiments

An initial investigation about the behaviour of one-sided selection methods was carried out using the *Breast Cancer* dataset from the UCI repository [4]. This dataset allows to visualise in three dimensions two cluster of cases that represent the classes *cancer=malignant* and *cancer=benign*. The following attributes were used as axes in the visualisation: *clump thickness*, *uniformity of cell size* and *bare nuclei*. Figure 5 shows the original dataset visualised through the Mineset tool from SGI.

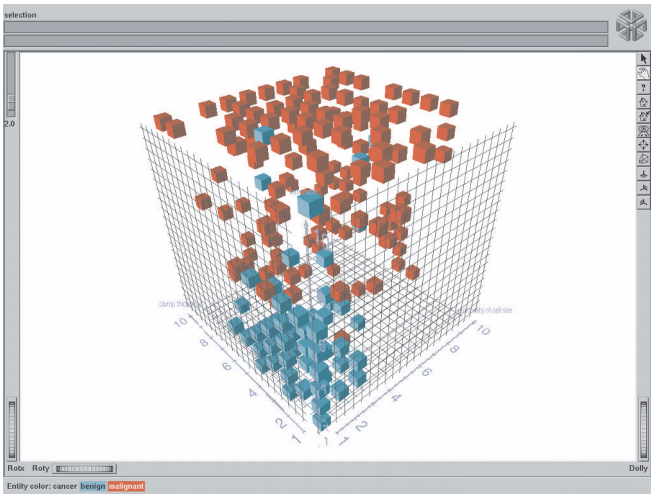

Fig. 5. Original Breast Cancer dataset.

First of all, cases belonging to the *cancer=benign* class (represented by light colour) that formed Tomek links were removed, since this class has some sparely distributed cases that might be noise. Afterwards, a consistent subset of the cases belonging to the *cancer=malignant* class (represented by dark colour) was found, since this class have some cases far from the border. Figure 6 shows the dataset after the application of one-sided selection methods.

7 Experimental Results

Some experiments were conducted to verify whether one-sided selection is able to improve the classification of the minority class in unbalanced datasets. The

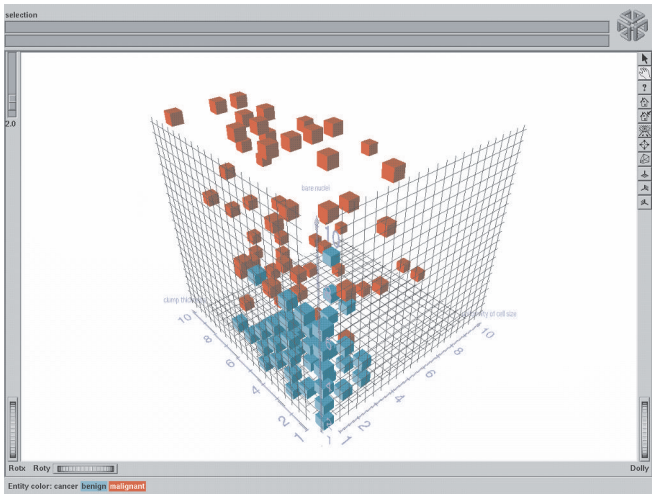

Fig. 6. Breast Cancer dataset without redundant, borderline and noisy cases.

C5.0 learning system and the *Hepatitis* dataset from UCI [4] were used for these experiments. The Hepatitis dataset has 155 cases with 123 cases (79,3%) belonging to the class *live* and 32 cases (20,6%) belonging to the class *die*. Also the Hepatitis dataset has both numeric and symbolic attributes whose distance was computed using the VDM metric. These experiments were performed with the aid of the AMPSAM environment [2, 3] to facilitate performance measures.

The hepatitis dataset is know by the Machine Learning community for its difficulty to produce good performance results. According to [8], very few learning systems obtained an accuracy higher than two percent points over the *baseline accuracy*, i.e., an accuracy two percent points over 79,3%.

C5.0 performance was measured on the original dataset with all cases (a); on the dataset without noisy and borderline cases removed by Tomek links (b); on the dataset without redundant cases removed by creating a consistent subset (c); on the dataset without noisy, borderline and redundant cases (d); and, finally, on the dataset without some majority class cases removed randomly (e). The error rates were measured using the 3-fold cross validation resampling method. The number of partitions $k = 3$ was chosen because there are very few minority class cases and higher k values might lead to high variance results. To confirm these results, 3-fold cross validation was applied 3 times. Since the results were similar, only the results of one of these experiments is shown in Table 2. The column N shows the number of examples in the training set (majority/minority class). Fn and Fp are the false negative and false positive rates, respectively. Columns $\delta(Fn)$ and $\delta(Fp)$ are the standard deviation for the false negative and false positive rates, respectively.

Comparing cases b and a using the standard hypothesis testing model the difference is significant for the minority class although not degrading significantly

	N	Fn	$\sigma(Fn)$	Fp	$\sigma(Fp)$
a	80/23	10%	5,8%	62%	5,1%
b	70/23	15%	4,0%	43%	11,7%
c	62/23	13%	8,6%	57%	11,7%
d	55/23	28%	4,5%	29%	7,9%
e	50/23	9%	2,6%	57%	17,4%

Table 2. Error rates and standard deviation.

for the majority class. Comparing cases d and a as well as d and e there is a significant improvement for the minority class but there is also a significant degradation for the majority class.

The results obtained suggest that one-sided selection can effectively decrease the error rate of the class *die* (minority class), specially when Tomek links are used (b and d). However, the random selection method (which does not use any heuristic) obtained some results comparable to the selection through consistent subsets method. Even though random selection does not apply any heuristic, it has the merit of removing the cases with equal probability. Probably, random selection is the method that causes the smallest change in the dataset distribution, comparing with the methods used in this work. There may exist other reasons for this effect, such as: most of the UCI datasets have already been carefully analysed before being made available in the repository. During these analysis, many borderline, noisy and redundant cases might have been removed, decreasing the effectiveness of the heuristics considered in this work. Another reason could be that minority class cases were left untouched, even though there might be noisy minority class cases. This decision was taken due to the necessity of keeping all the few cases belonging to the minority class. However, noise present in the minority class cases can reduce the classification accuracy. In this work, only majority class cases forming Tomek links were removed. However, Tomek links might be related to noisy cases from the minority class. By only removing majority class cases forming Tomek links, minority class noisy cases may not be removed. Moreover, important majority class cases may be removed. Since minority class cases are precious, an improvement to the current method is to distinguish between noisy and borderline cases. Unfortunately, Tomek links do not offer a safe method to distinguish such types of cases and other methods should be investigated.

8 Conclusions

Datasets with a large difference between the number of cases belonging to each class are common in Machine Learning. For those datasets, it may be trivial to produce a classifier that performs very well on the majority class cases. However, accurately classify the cases of the minority class is much more difficult to achieve. Several practitioners have been solving this problem by replicating (sometimes with little noise) minority class cases. However, this technique poten-

tially introduces bias in the dataset. One-sided selection does a careful removal of majority class cases, aiming to balance the number of cases of each class. This work shows some results that seem to suggest the effectiveness of this technique which is actually being implemented as part of a computational system for data pre-processing. In future works, we will look for new heuristics to discriminate between borderline and noisy cases in order to select and remove noisy minority class cases. Also, we will look for new heuristics to select majority class cases.

Acknowledgements The authors wish to thank an anonymous referee of MICAI-2000 for his/her helpful comments on this paper. This research was partly supported by FINEP and Silicon Graphics Brasil.

References

1. Barnard, E., Cole, R.A., Hou, L.: Location and Classification of Plosive Constants Using Expert Knowledge and Neural Nets Classifiers. Journal of the Acoustical Society of America, **84** Supp 1:S60 (1988).
2. Batista, G.E.A.P.A.; Monard, M.C.: A Computational Environment to Measure Machine Learning Systems Performance (in Portuguese). Proceedings I ENIA (1997) 41–45.
3. Batista, G.E.A.P.A.; Monard, M.C.: An Implementation Description of the Statistical Methods used in the AMPSAM Environment (in Portuguese). Technical Report 68 ICMC-USP ISSN-0103-2569 (1998).
4. Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine Learning Databases. Irvine, CA: University of California, Department of Information and Computer Science. http://www.ics.uci.edu/ mlearn/MLRepository.html.
5. Chan, P.K., Stolfo, S.J.: Learning with Non-uniform Class and Cost Distributions: Effects and a Distributed Multi-Classifier Approach.KDD-98 Workshop on Distributed Data Mining (1998) 1–9.
6. Cost, S., Salzberg, S.:A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features.Machine Learning **10** (1993) 57–78.
7. Hart, P.E.:The Condensed Nearest Neighbor Rule.IEEE Transactions on Information Theory **14** (1968) 515–516.
8. Holte, C.R.:Very Simple Classification Rules Perform Well on Most Commonly Used Datasets.Machine Learning **11** (1993) 63–91.
9. Kubat, M., Matwin, S.:Addressing the Course of Imbalanced Training Sets: One-Sided Selection. Proceedings of the 14th International Conference on Machine Learning - ICML'97, Morgan Kaufmann (1997) 179–186.
10. Lawrence, S., Burns, I., Back, A., Tsoi, A.C., Giles, C.L.: Neural Network Classification and Prior Class Probabilities.Tricks of the trade, Lecture Notes in Computer Science State-of-the-art Surveys, G. Orr, K.R. Müller, R. Caruana (editors), Springer Verlag (1998) 299–314.
11. Lewis, D., Catlett, J.: Heterogeneous Uncertainty Sampling for Supervised Learning.Proceedings of the 11th International Conference on Machine Learning - ICML' 94, Morgan Kaufmann (1994) 148–156.
12. Quinlan, J.R.:C4.5 Programs for Machine Learning.Morgan Kaufmann Publishers, CA (1988).
13. Stanfill, C., Waltz, D.: Toward Memory-Based Reasoning.Communications of the ACM, **29** (1986) 1213–1228.

14. Stolfo, S.J., Fan, D.W., Lee, W., Prodromidis, A.L., Chan, P.K.: Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results.Proceedings of AAAI-97 Workshop on AI Methods in Fraud and Risk Management (1997).
15. Tomek, I. :Two Modifications of CNN. IEEE Transactions on Systems Man and Communications **6** (1976) 769–772.