Int. J. Appl. Math. Comput. Sci., 2016, Vol. 26, No. 1, 191–201
DOI: 10.1515/amcs-2016-0013

USING THE ONE–VERSUS–REST STRATEGY WITH SAMPLES BALANCING TO IMPROVE PAIRWISE COUPLING CLASSIFICATION

WIESŁAW CHMIELNICKI [a,*], KATARZYNA STĄPOR [b]

[a]Faculty of Physics, Astronomy and Applied Computer Science
Jagiellonian University, ul. prof. Stanisława Lojasiewicza 11, 30-348 Kraków, Poland
e-mail: wieslaw.chmielnicki@uj.edu.pl

[b]Institute of Computer Science
Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland
e-mail: katarzyna.stapor@polsl.pl

The simplest classification task is to divide a set of objects into two classes, but most of the problems we find in real life applications are multi-class. There are many methods of decomposing such a task into a set of smaller classification problems involving two classes only. Among the methods, pairwise coupling proposed by Hastie and Tibshirani (1998) is one of the best known. Its principle is to separate each pair of classes ignoring the remaining ones. Then all objects are tested against these classifiers and a voting scheme is applied using pairwise class probability estimates in a joint probability estimate for all classes. A closer look at the pairwise strategy shows the problem which impacts the final result. Each binary classifier votes for each object even if it does not belong to one of the two classes which it is trained on. This problem is addressed in our strategy. We propose to use additional classifiers to select the objects which will be considered by the pairwise classifiers. A similar solution was proposed by Moreira and Mayoraz (1998), but they use classifiers which are biased according to imbalance in the number of samples representing classes.

Keywords: pairwise coupling, multi-class classification, problem decomposition, support vector machines.

1. Introduction

Classification tasks are widely used in real-world applications. Most of them are classification problems that involve more than two classes. We call them multi-class problems. There are many methods of decomposing such a task into the set of the smaller classification problems involving two classes only. Benefits obtained from the decomposition of the multi-class task have been addressed by many authors (e.g., Allwein *et al.*, 2001; Kahsay *et al.*, 2005; Ou and Murphey, 2006; Krzysko and Wolynski, 2009; Saez *et al.*, 2012).

Among the methods of decomposition, pairwise coupling proposed by Hastie and Tibshirani (1998) is one of the best known. In general, its principle is to separate each pair of classes ignoring the remaining ones. In this way a number of binary classifiers are trained between all possible pairs of classes. The multi-class problem with K classes creates $K(K-1)/2$ binary sub-problems and the corresponding binary classifiers.

Then all the objects represented by the feature vectors are tested against these binary classifiers, and in the next step a voting scheme is used. Friedman (1996) proposed a max-voting scheme, which means that the object with the maximum number of votes is classified as the correct class. Hastie and Tibshirani (1998) suggested that it can be improved by using pairwise class probability estimates in a joint probability estimate for all classes.

A closer look at the pairwise strategy shows the problem which impacts the final result of the combined classifier. Each binary classifier votes for each object even if it does not belong to one of the two classes which it is trained on. So we use the class probability estimates produced by this classifier even if the object belongs to the class which the classifier is not aware of, i.e., objects representing this class are not present in the training data set of the classifier.

*Corresponding author

This problem is addressed in our strategy. In our solution, additional correcting classifiers are used to select the objects which will be considered by the pairwise classifiers. A similar solution was proposed by Moreira and Mayoraz (1998), but they use classifiers which biased according to the imbalance in the number of objects representing the classes.

The proposed solution was tested on several databases using two different classifiers. We employed four real life databases: MNIST (modified NIST) (LeCun *et al.*, 2014), the Gesture database (Glomb *et al.*, 2011), Proteins (Ding and Dubchak, 2001), Gestures II (database of 32 gestures created by the authors of this paper) and six other databases from the UCI Machine Learning Repository (UCIMLR, 2014). The obtained results show that our strategy outperforms not only the original pairwise coupling algorithm but also the solution proposed by Moreira and Mayoraz (1998). The difference is more significant when the number of classes in the problem is growing.

2. Related work

There are many methods of decomposition of multi-class problems into a set of the binary classification problems such as the OVR (one-versus-rest) and OVO (one-versus-one) strategies, DAG (directed acyclic graph) and ADAG (adaptive directed acyclic graph) methods (Platt *et al.*, 2000; Kijsirikul and Ussivakul, 2002), the BDT (binary decision tree) approach (Fei and Liu, 2006), the DB2 method (Vural and Dy, 2004), PWC (pairwise coupling) (Hastie and Tibshirani, 1998) or ECOCs (error-correcting output codes) (Dietterich and Bakiri, 1995).

Additionally, some interesting reviews considering this topic can be found in the works of Lorena *et al.* (2008) or Krzysko and Wolynski (2009). We can also look at the problem of decomposition from the efficiency point of view (Chmielnicki *et al.*, 2012), or we can investigate how the problem properties can be employed for the construction of the decomposition scheme (Lorena and Carvalho, 2010).

Another approach based on an ensemble of binary predictors is presented by Galar *et al.* (2011). This paper provides a study on the one-versus-one and one-versus-rest methods, with special attention on the final step of the ensembles; the combination of the outputs of the binary classifiers. The dynamic classifier selection strategy for the one-versus-one scheme that tries to avoid non-competent classifiers is addressed by Galar *et al.* (2013).

Worth mentioning is also the one class classifiers (OCC) approach. For example, we can propose building an ensemble of one-class classifiers based on the clustering of the target class (Krawczyk *et al.*, 2014). The main advantage of such a method is that the combined classifiers trained on the basis of clusters allow us to exploit individual classifier strengths.

One of the best known and widely used methods of decomposition is one-versus-one strategy, where the input vector x is presented to the binary classifiers trained against each pair of the classes. We can assume that each classifier discriminates between class ω_i and class ω_j and computes the estimate $\hat{p}_{ij}$ of the probability

$$p_{ij} = P(x \in \omega_i | x, x \in \omega_i \cup \omega_j). \tag{1}$$

Then the classification rule is defined as

$$\arg \max_{1 \le i \le K} \sum_{j \ne i} I(\hat{p}_{ij}), \tag{2}$$

where K is the number of the classes and $I(\hat{p}_{ij})$ is defined as

$$I(\hat{p}_{ij}) = \begin{cases} 1, & \hat{p}_{ij} > 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

This approach was proposed by Friedman (1996) and we call it the max-voting scheme. Another approach was suggested by Hastie and Tibshirani (1998) as well as Moreira and Mayoraz (1998). We can take into consideration that the outputs $\hat{p}_{ij}$ of the binary classifiers represent the class probabilities. Consequently, these values can be used as the estimates $\hat{p}_i$ of *a posteriori* probabilities

$$p_i = P(x \in \omega_i | x), \tag{4}$$

Assuming that we have a square matrix $K \times K$ of $\hat{p}_{ij}$'s for $i, j = 1 \dots K$ and $\hat{p}_{ji} = 1 - \hat{p}_{ij}$, we can calculate the values of $\hat{p}_i$'s as

$$\hat{p}_i = \frac{2}{K(K-1)} \sum_{j \ne i} \sigma(\hat{p}_{ij}), \tag{5}$$

for $i = 1, \dots, K$, and then we can use the classification rule

$$\arg \max_{1 \le i \le K} \hat{p}_i, \tag{6}$$

where σ takes the form a threshold function at 0.5 for the max-voting scheme and the identity function for the solution proposed by Hastie and Tibshirani (1998). Some other σ functions are considered by Moreira and Mayoraz (1998).

If we look closer at the PWC decomposition scheme, we will see that in all approaches we are using values of $\sigma(\hat{p}_{ij})$ for a given vector x which belongs neither to the class ω_i nor to ω_j. Looking at (5), we see that the estimation of p_i takes into account all classifiers even if they are not trained on the samples of the class to which x belongs to.

For example, let us consider the classifier which has been trained on the samples of ω_i and ω_j classes.

Accordingly, if x belongs to some other class (let us say, k, $k \neq i$ and $k \neq j$), then $\hat{p}_{ik}$ and $\hat{p}_{kj}$ are completely irrelevant because the referred classifier has no competence to deal with the class ω_k. There were no samples of the class ω_k in its training data set.

A procedure to overcome this problem was proposed by Moreira and Mayoraz (1998), which consists in training additional correcting classifiers separating the classes i and j from all the other classes. These classifiers produce the outputs $\hat{q}_{ij}$, which provide us with an estimate of the probability that sample x belongs to the class i or to the class j. Therefore, we can modify (5), which now becomes

$$\tilde{p}_i = \frac{2}{K(K-1)} \sum_{j \neq i} \sigma(\hat{p}_{ij}) \hat{q}_{ij}. \tag{7}$$

The use of these classifiers should cause that the incompetent classifiers have no significance and improve the quality of the estimation $\hat{p}_i$. Another approach to correcting classifiers using the weights (produced by a different classifier) was proposed by Chmielnicki and Stąpor (2010). We can also consider the neighborhood of each instance to decide whether a classifier is competent or not (Galar *et al.*, 2013).

Moreira and Mayoraz (1998) proved that the correcting procedure they proposed is able to improve the performance of the decomposition scheme. However, this improvement is achieved at the cost of having twice as many classifiers as in the standard PWC algorithm, because we need one correcting classifier for each pair of the classes. The authors point out that this problem can be eased by distributing these classifiers, especially using multi-core and multi-processor machines.

They compared PWC methods using different σ functions with PWC-CC methods including correcting classifiers on several databases. The results show that this solution decreased misclassification errors on all the tested data sets.

3. Comparison of the OVO and OVR methods

Pairwise coupling is using the OVO (one-versus-one) strategy employing binary classifiers between each pair of the classes. The correcting classifiers introduced by Moreira and Mayoraz (1998) use the OVR (one-versus-rest) strategy. Both the strategies have their advantages and disadvantages which may impact the final result of the combined classifier. The strategies will be shortly discussed in this section.

When we use the OVO strategy, we have to train a set of $K(K-1)/2$ binary classifiers between each pair of the classes. Then all the samples representing all classes are tested against these classifiers which vote for each sample. This brings us to the problem of incompetent classifiers and votes that should be ignored, which was mentioned in the previous section.

This problem is clearly visible in Fig. 1. The *2 vs 8* classifier is used to test the samples of all classes from 0 to 9. We can see, for example, that all the samples of class 4 are classified as class 2. On the other hand, some samples of class 6 are classified as class 8 but some other as class 2, which is even worse especially when we are using the max-voting scheme.

Another problem can be seen when the number of classes increases. The number of binary classifiers rises quadratically and all the samples have to be tested against each classifier during the testing phase. For example, 1000 classes mean about half a million of binary classifiers. There are several methods to deal with the issue. Some solutions addressing this problem were proposed by Chmielnicki and Stąpor (2012).

The OVR strategy uses samples of all the classes to train each binary classifier. However, the samples from one distinguished class are treated as the class *one*, ω_1, and all the other samples are considered to belong to the class *rest*, ω_r. When we are using the Moreira and Mayoraz correcting classifiers, we treat the samples from the two classes i and j as the class ω_1 and all the others as the class ω_r.

Compared with the OVO strategy, the number of binary classifiers which we have to train is quite small. We need K binary classifiers only, and we see that this number increases linearly with the number of classes. We should also notice that we have many more samples in the training data set at the training phase, especially when the number of classes is large. This can impact the training time. However, usually the time of the training phase is much less important that the time of the testing phase.

When the number of classes increases, yet another problem can be seen. As has been stated earlier in this

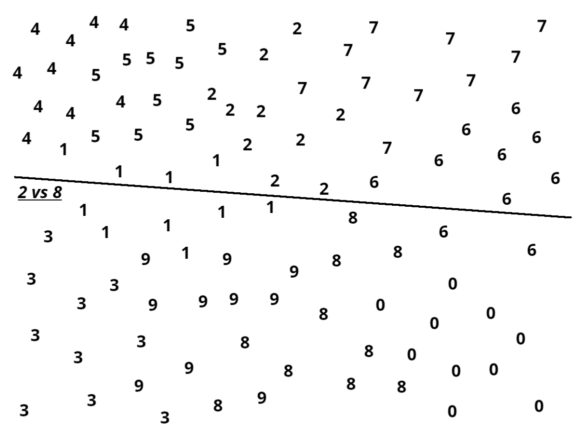

Fig. 1. One-versus-one approach.

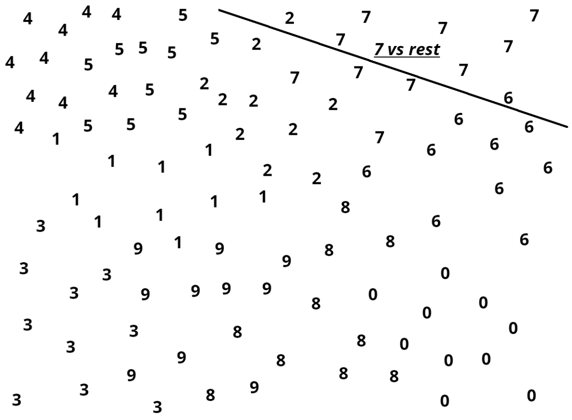

Fig. 2. One-versus-rest approach.

section, almost all the samples representing the classes except the one distinguished class are treated as one big class. This causes the problem of overrepresenting the *rest* class. Therefore, the result of these binary classifiers could be very biased. For example, if we have 1000 classes and the classes are represented by the same number of samples, then we will have 999 times more samples of the *rest* class than the samples of the *one* class. In this problem, if a learning algorithm classifies all the samples as the majority class, it achieves a very high recognition ratio, i.e., 99.9%.

We can see this problem in Fig. 2. Three samples of the 7 class were misclassified because the class *rest* is overrepresented. When the number of classes increases, the problem is much worse.

The issue mentioned above is widely known and was addressed in several papers (e.g., Chawla *et al.*, 2002; Liu *et al.*, 2008; He and Garcia, 2009; Cateni *et al.*, 2014; Beyan and Fisher, 2015). Generally, there are two popular methods dealing with class-imbalance problems: over-sampling the minority class and under-sampling the majority class.

In the former approach we create "synthetic" samples representing the minority class or we duplicate real data entries. Under-sampling is a method which uses only a subset of samples from the majority class. The main deficiency of this approach is that many majority class samples are ignored.

4. Proposed method

As we stated in the previous section, one of the weaknesses of the Moreira and Mayoraz approach is the number of correcting classifiers. Another weakness can be noticed when we look at Fig. 3. We use the OVR scheme for every possible class, treating samples from two different classes as samples of the same class. If the classes are similar, the results can be quite good (see the classifier *4,5 vs rest* in Fig. 3 but we are training correcting classifiers for all possible pair of classes. For example, if we look at *0,7 vs rest* classifier, the results are not so encouraging.

We can notice that instead of using *0,7 vs rest* we can use the *0 vs rest* and *7 vs rest* classifiers. The results of these classifiers will be usually much better. However, we need the values of $\hat{q}_{ij}$ to evaluate (7). We can obtain these values as

$$\tilde{q}_{ij} = \max(\tilde{p}_i, \tilde{p}_j), \tag{8}$$

where $\tilde{p}_i$ and $\tilde{p}_j$ are the estimates that the sample x belongs the to the class i or the class j, respectively.

This approach decreases the number of correcting classifiers needed from $K(K-1)/2$ to K and we do not mix samples from two different classes into one. Of course, the problem of the overrepresenting *one* class in the OVR strategy is even more visible in this solution, but we will try to deal with it in the next step.

The main problem visible in the solution proposed by Moreira and Mayoraz (1992) is that the number of samples of the *one* class is much smaller than that of samples of the class *rest*. It will be even more serious if we use the solution proposed in our paper. Moreover, the problem is more and more visible when the number of classes grows. As a consequence, the result of the correcting classifier can be very biased.

The problem with the imbalance of the number of samples representing classes is visible in many applications. It has been discussed by many authors (one of the interesting works is that of He and Garcia (2009)). It usually occurs when we have more samples of one class than of the others. In such cases, classifiers tend to be overwhelmed by the large class and ignore the small one. They tend to produce high predictive accuracy for the majority class but poor accuracy for the minority class.

A number of solutions to class-imbalance problems have been proposed both at the data and algorithmic

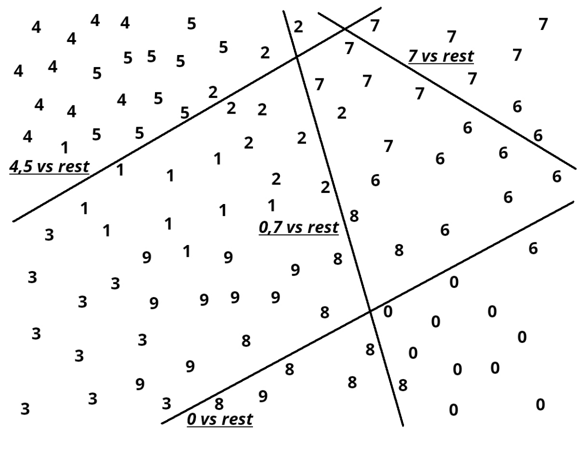

Fig. 3. Correcting classifiers.

levels. As we mentioned in the previous section, two of them: over and under-sampling are most popular. However, over-sampling increases the training data set size and thus requires longer training time. Furthermore, it tends to lead to overfitting since it repeats minority class samples (Chawla *et al.*, 2002). Consequently, in our solution we focus on the method which is a kind of under-sampling.

In our first approach to the problem we suggest to use the subset of the *rest* data set by sampling the whole set to balance the number of samples in *one* set N_{one} and in the *rest* set N_{rest}, see Fig. 4. However, this solution will not work well because of the random choice of the samples. Consequently, to improve the solution, we can use the method resembling bagging (Breiman, 1996). We can draw M different data subsets from the *rest* data set. Then we can use the average result from the M classifiers which have been taught on these training data subsets, i.e.,

$$\hat{q}_i = \frac{1}{M}\sum_{j=1}^{M} \hat{q}_{ij}. \tag{9}$$

A weak point of this procedure is that the samples of some classes will not be present in some *rest* data subsets. There is even a possibility that some particular *rest* data subset will be constituted from samples of one class only. The problem is more visible when $N_{\text{one}} \leq K$. This will impact the result of such a classifier. To avoid this situation, we can change the drawing procedure to look for classes with the same numbers.

Algorithm 1. Building the *rest* data subset.

Require: N_{one}, rest_dataset
rest_subset := $\emptyset$
while $(N_{\text{one}} > 0)$ **do**
  sample, class := GetSample(rest_dataset)
  if not IsPresent(class, rest_subset) **then**
    $N_{\text{one}} := N_{\text{one}} - 1$
    rest_subset := rest_subset + sample
  end if
end while
return rest_subset

The procedure of building the *rest* data subset is described in Algorithm 1. Once more, observe that when $N_{\text{one}} > K$ we have to change the *IsPresent* function. Now, for the first K samples it should check if the sample of the class *class* is present in *rest_subset*, but for the next K it should check if the sample of the class *class* is represented in *rest_subset* at least once and so on.

However, if $N_{\text{one}} < K$, then even using the procedure described in Algorithm 1 we cannot avoid the situation that there are some classes which are not represented in the *rest* data subset. To solve this problem, we allow a small imbalance between the number of samples for the sake of representing of all the classes in the *rest* data subset. In our experiments we used the formula below to set the number of samples in the *rest* data subset,

$$N_{\text{rest}} = \min\{2(K-1), N_{\text{one}}\}, \tag{10}$$

where N_{rest} is the number of samples in the *rest* data subset, N_{one} is the number of samples representing *one* class and K is the number of all classes. This guarantees us that at least two samples of each class are present in the training set for the *rest* class.

The above formula offers some trade-off between balancing the data sets and the problem that every class should be represented in the training data set. As we can see in Fig. 5, the results using this strategy overcome the two others.

All the correcting classifiers produce the values of the probabilities $\hat{q}_{ij}$ used in (7). Evaluating this formula requires testing every sample against each of the $K(K-1)/2$ OVO classifiers to get the values of $\hat{p}_{ij}$. It is a very

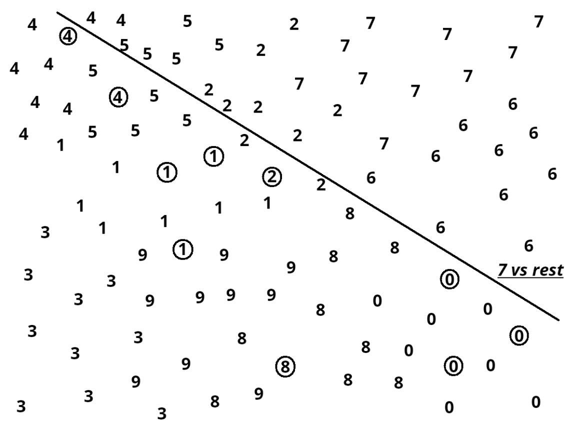

Fig. 4. Approach with strict balancing.

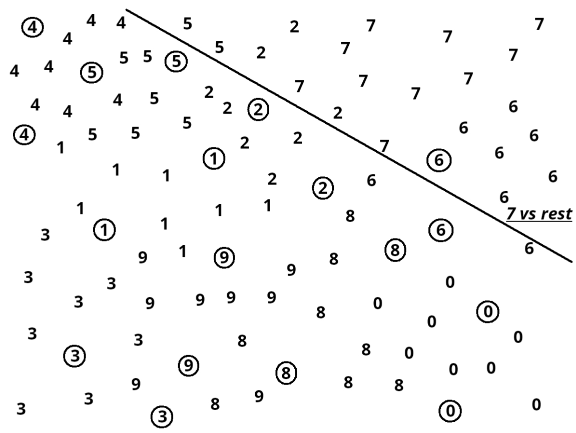

Fig. 5. Approach with soft balancing.

expensive operation, especially if the number of classes is very large.

It can be easily noticed that some of the probabilities $\hat{q}_{ij}$ are quite small, so it makes no sense to test a sample with a small value of $\hat{q}_{ij}$. By introducing a threshold for the value of $\hat{q}_{ij}$, we can speed up our algorithm. In our experiments we tested the samples with $\hat{q}_{ij} > 0.25$ only, and the result of the final classifier did not change.

Further experiments can be carried out to test how the value of this threshold impacts both classifier speed and accuracy. We can see that for the maximum sensible value of the threshold $\hat{q}_{ij} \geq 0.5$ we have to test $2N/K$ samples in the most optimistic case (assuming that every class is represented by the same number of samples and that each of the OVR classifiers has 100% accuracy).

5. Results of the experiments

Several experiments were conducted to test the proposed methods. Two different classifiers were used: the support vector machine (SVM), which represents the generative approach to the classification task, and linear discrimination analysis (LDA), which is a discriminative classifier. We used these classifiers because we wanted to check if our solution could be applied with these two kinds of approaches. The characteristics of these classifiers differ in several respects. For a more detailed discussion, see the work of Liu and Fujisava (2005). Both the classifiers are briefly described in the following paragraphs.

The support vector machine is a well-known large margin classifier proposed by Vapnik (1995). The basic concept behind the SVM classifier is to find an optimal separating hyperplane, which separates two classes. The decision function of the binary SVM is

$$f(x) = \mathrm{sign}\Big(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\Big), \qquad (11)$$

where $0 \leq \alpha_i \leq C, i = 1, 2, \ldots, N$, are nonnegative Lagrange multipliers, C is a cost parameter which controls the trade-off between allowing training errors and forcing rigid margins, x_i are the support vectors and $K(x_i, x)$ is the kernel function.

Quadratic discriminant analysis (QDA) models the likelihood of a class as a Gaussian distribution and then uses the posterior distributions to estimate the class for a given test vector. This approach leads to the discriminant function

$$d_k(x) = (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \ln |\Sigma_k| - 2 \ln \pi_k \;, \quad (12)$$

where x is the test vector, μ_k is the mean vector, Σ_k the covariance matrix and p_k is the prior probability of the class k. The Gaussian parameters for each class can be estimated from the training data set, so the values of Σ_k and μ_k are replaced in the formula (12) by its estimates $\hat{\Sigma}_k$ and $\hat{\mu}_k$.

However, when the number of training samples N is small compared with the number of dimensions of the training vector, the covariance estimation can be ill-posed. The approach to resolve the ill-posed estimation is to replace $\hat{\Sigma}_k$ by the pooled covariance matrix, i.e.,

$$\hat{\Sigma} = \frac{1}{N - K} \sum_{k=1}^{K} \sum_{y_i = k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T, \qquad (13)$$

which brings us to linear discriminant analysis with the decision function as

$$d_k(x) = x^T \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + \ln \pi_k. \qquad (14)$$

We have to employ several databases with different characteristics to test our solution. Some of the databases can be found in the UCI Machine Learning Repository (UCIMLR, 2014). We also used the databases MNIST and Gestures, created at the Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences (Glomb *et al.*, 2011). The databases Leafs II and Leafs III are in fact the same database but used with different feature vectors (based on shapes—Leafs II, and based on margins—Leafs III). In Table 1 we show the number of classes and the size of the feature vector for all databases used.

Table 1. Databases used in the experiments.

Name	Classes	Samples	Features
MNIST	10	70 000	102
Activities	19	9 120	45
Gestures	22	1 320	256
ISOLET	26	7 797	617
Proteins	27	698	126
Gestures II	32	640	512
Leafs	36	340	13
ACRS	50	1 500	10 000
AusLan	95	2565	88
Leafs II	100	1 600	64
Leafs III	100	1 600	64

On each database, four algorithms were tested, i.e., one-versus-one (OVO), pairwise coupling (PWC), pairwise coupling with corrected classifiers (PWC-CC), proposed by Moreira and Mayoraz (1998), and our algorithm—pairwise coupling with samples balancing (PWC-B). All these algorithms were tested using the SVM and LDA classifiers. The results (the average recognition ratios from the k-crossvalidation procedure) are presented in Tables 2 and 3.

The LDA classifier was implemented by the authors. The implementation of the SVM classifier used in the experiments is from the work of Chang and Lin (2001).

Figure 6 shows the results obtained by the correcting classifiers using different methods of balancing. The first method, **Random**, just draws N_{one} samples from the *rest* data set. In the next approach, **RCB** (random with class balancing), we use the *class aware method* described in the Algorithm 1, and finally we apply our ultimate solution, **M-RCB** (modified random with class balancing) to ensure that at least two samples of each class are present in the *rest* data set.

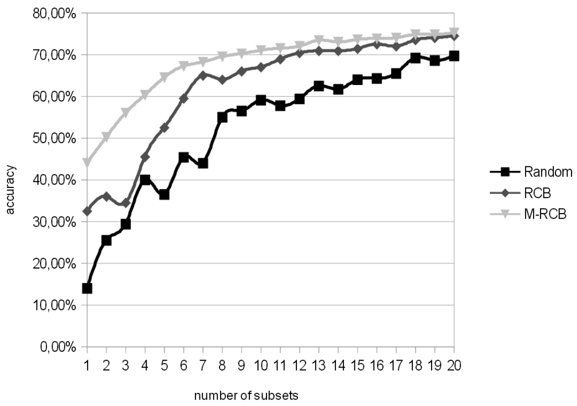

Fig. 6. Correlation between accuracy and M—the number of data subsets.

In our solution we propose three random procedures to balance the number of samples in the *one* and the *rest* data sets, which should improve the recognition ratios achieved by the correcting classifiers. We also use a procedure of selecting M data subsets from the *rest* data set to obtain better results. The relationship between the number of data subsets M and the average recognition ratio obtained by binary classifiers is presented in Fig. 6.

We can notice from the figure that the accuracy is not monotone increasing when the number of subsets is growing. This is not surprising because we are using a randomized procedure to generate the *rest* data subsets. Some of these data sets are very poor. For example, we can imagine the situation when we draw to the *rest* data set samples representing one class only.

However, when we are using the **RCB** method, this situation is not possible but still we may obtain the *rest* data subset, which does not contain any sample representing one or more classes. This problem is even more visible when the number of classes is large and the number of samples representing each class is small (it always happens when $N_{\text{one}} < K$).

Finally, the last approach, **M-RCB**, guarantees that all the classes are represented at the cost of having a slightly imbalanced data sets. We see that this approach gives us the best accuracy but also the accuracy, is increasing with the number of the generated data subsets.

In our experiments we just start from training $M \times K$ binary correcting classifiers, and then we calculate $\hat{q}_{ij}$ probabilities for each sample from the testing data set. Finally, we apply the standard PWC algorithm using these probabilities, but each OVO classifier is running only against samples which have $\hat{q}_{ij} > 0.25$.

The procedure of k-crossvalidation was used to avoid biased results. We use $k = 10$ in our experiments. Only the average value of the k-crossvalidation is shown in the tables. We can observe that our solution overcomes all other algorithms on all databases no matter which classifier is used. Only the results obtained on the MNIST database are almost the same.

The results of the PWC-B algorithm are better than those of PWC by 1.2 to 2.5% for the LDA classifier and by 0.7 to 2.6 for the SVM classifier (we neglect the results for the MNIST database, which will be discussed later in the next section). When we compare the results of the PWC-B versus the PWC-CC algorithm, we obtain 0.6 to 3.1 and 0.5 to 3.2, respectively. The question is if this difference is statistically significant.

There are many methods described in the literature which deal with the comparison of classfiers, starting from the most cited (Dietterich, 1998), recommending the $5 \times 2cv$ t-test, while Nadeau and Bengio (2003) propose the corrected resampled t-test that adjusts the variance

Table 2. Results using the LDA classifier.

DB name	OVO	PWC	PWC-CC	PWC-B
MNIST	98.7%	98.8%	98.7%	98.8%
Activities	92.2%	93.5%	93.9%	94.8%
Gestures	86.2%	86.5%	87.1%	87.7%
ISOLET	94.1%	94.2%	94.6%	96.0%
Proteins	56.1%	56.3%	56.5%	58.1%
Gestures II	58.4%	59.5%	59.5%	60.9%
Leafs	75.2%	76.1%	76.0%	78.6%
ACRS	65.4%	65.7%	65.5%	67.7%
AusLan	85.2%	85.8%	86.3%	88.4%
Leafs II	69.1%	71.2%	70.7%	73.4%
Leafs III	84.5%	85.7%	85.0%	88.1%

Table 3. Results of testing the accuracy of the SVM classifier.

DB name	OVO	PWC	PWC-CC	PWC-B
MNIST	99.0%	99.1%	99.1%	99.1%
Activities	95.1%	95.8%	96.2%	96.9%
Gestures	81.2%	81.8%	82.1%	82.9%
ISOLET	96.3%	96.4%	96.6%	97.1%
Proteins	57.2%	58.0%	57.9%	58.9%
Gestures II	60.2%	60.5%	60.3%	61.7%
Leafs	79.1%	79.5%	79.4%	80.7%
ACRS	73.4%	73.7%	73.1%	75.6%
AusLan	87.2%	87.4%	87.7%	90.5%
Leafs II	72.6%	74.5%	74.2%	76.9%
Leafs III	85.6%	86.4%	85.8%	89.0%

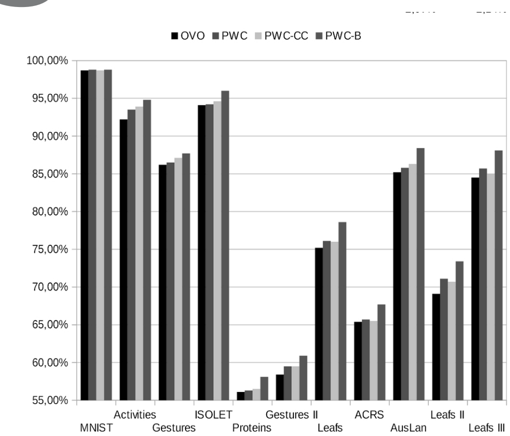

Fig. 7. Results obtained using the LDA classifier.

based on the overlaps between subsets of examples. However, the most comprehensive study on this subject was prepared by Demsar (2006). He recommended to use the Wilcoxon (1945) signed-ranks test for comparisons of two classifiers.

We tested PWC-B versus the original PWC algorithm and PWC-B versus PWC-CC using the Wilcoxon signed-ranks test. The results show that in both cases the difference is statistically significant at the significance level equal to 0.05.

Additionally, in the next section, we present a more detailed comparison of the proposed classifiers using the Iman and Davenport test with the Nemenyi post hoc analysis.

6. Statistical comparison of the classifiers

As the last step of our experiments, we test the null hypothesis that all tested classifiers (i.e., PWC-B, PWC, PWC-CC and OVO) perform the same and the observed differences are merely random. We used the Iman and Davenport test (Iman and Davenport, 1980), which is a nonparametric equivalent of ANOVA.

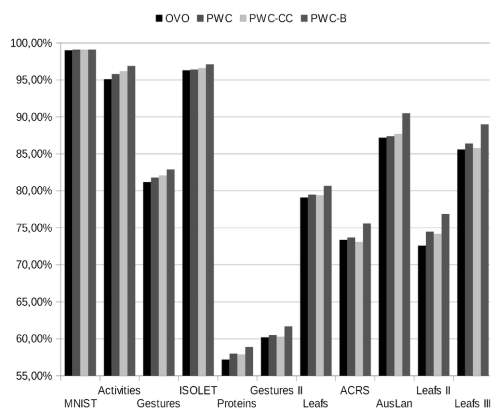

Fig. 8. Results obtained using the SVM classifier.

Let R_{ij} be the rank of the j-th of K classifiers on the i-th of N data sets. The test compares the mean ranks of the classifiers and it is based on the statistic

$$F_f = \frac{(N-1)\chi_f^2}{N(K-1)-\chi_f^2}, \tag{15}$$

where

$$\chi_f^2 = \frac{12N}{K(K+1)} \sum_{i=1}^{K} R_i^2 - 3N(K+1) \tag{16}$$

is the Friedman statistic which is distributed according to the F distribution with $K-1$ and $(K-1)(N-1)$ degrees of freedom and

$$R_i = \frac{1}{N} \sum_{i=1}^{N} R_{ij}. \tag{17}$$

In our case, the p-value of the test statistic is equal to $p = 9.9366 \times 10^{-12}$ for SVM classifiers and $p = 1.9462 \times 10^{-13}$ for LDA classifiers. We see that the null hypothesis that all classifiers give the same results is rejected (as the p-value is less than the significance level).

Hence in the next step we can use the Nemenyi post hoc test (Nemenyi, 1963), in which all classifiers are compared to each other. The performance of two classifiers is significantly different at the significance level α if the corresponding average ranks differ by at least the critical difference (CD):

$$|R_i - R_j| > CD = q(\alpha, K, \infty)\Big(\frac{K(K+1)}{12N}\Big)^{1/2}, \tag{18}$$

where $i = 1, \dots K-1, j = i+1, \dots, K$, and where the critical values of $q(\alpha, K, \infty)$ are based on the Studentized range statistic and can be found, for example, in the work of Hollander and Wolfe (1973).

In our cases, for $\alpha = 0.1$, $K = 4$, $N = 11$, the right-hand side of the inequality (18), i.e., the critical distance CD, is equal to 1.3. The results of the multiple comparisons are presented graphically for SVM and LDA in Figs. 9 and 10, respectively.

Those classifiers connected by a vertical line have average ranks that are not significantly different from each other. Those groups are identified using the average rank of a model $\pm$ the critical distance.

The mean ranks of the model for the classifiers PWC-B, PWC, PWC-CC, OVO are

$$\begin{array}{ll} 3.9091, 2.3636, 2.6364, 1.0909 & \text{for SVM,} \\ 3.9545, 2.4545, 2.5455, 1.0455 & \text{for LDA.} \end{array}$$

(Classifiers are listed in accordance with their ranking.) We obtained three disjoint, homogenous groups of classifiers (Figs. 9 and 10):

PWC-B, (PWC, PWC-CC), OVO.

We see that the best classifier is contained in the first group, which is composed of only one classifier, the PWC-B one.

7. Conclusions

The problem with imbalanced data sets is intrinsic when we are using the one-versus-rest approach. Moreover, it grows with the number of classes used. It impacts the results of correcting classifiers and therefore the final result of the experiment. In the first step we proposed the method (a kind of under-sampling) which requires that the numbers of the classes be the same in the *one* and the *rest* data sets, which improves the result but causes another problem, with some classes being not represented in the *rest* data set.

Therefore, in the next step we suggest to introduce some trade-off between balancing data sets and the goal that all the classes should be represented in the *rest* data set. This solution was tested against several databases. These represent various domains of science and technology and as we can see from Table 1 they have very different characteristics. This means they have different numbers of classes, and samples, and different sizes of feature vectors.

The results obtained from the experiments are encouraging. They show that our algorithm overcomes the other ones almost on all tested databases. Only the results on the MNIST database are the same. This database consists of 10 classes only, which means that the problem of the imbalanced data sets is almost not visible in this case. Additionally, our solution addresses the problem of incompetent binary classifiers used in the PWC algorithm. The problem which is almost not seen in this particular database is that binary classifiers obtain recognition ratios over 99.6%.

The results and the analysis of the proposed method suggest that it should perform better as the number of classes is increasing, which means that the problem of imbalanced data sets is also more serious. Considering the fact that the average number of classes in the databases from the UCI Machine Learning Repository increases from 5.6 in the years 1988–1992 to 35.3 in the years 2008–2012, this is a very important result.

Obviously, we lose some effectiveness due to the usage of correcting classifiers, but we neutralize this effect using a threshold based on $\hat{q}_{ij}$ values, reducing the number of samples we have to test against each from $K(K-1)/2$ OVO binary classifiers. The experiments show that the proposed solution is as efficient as the original PWC method.

Some future experiments in this area may be interesting in two different aspects. We can consider how much the *rest* data set might be imbalanced to get better accuracy and what threshold value for correcting classifiers should be used to improve the speed of the combined algorithm without losing accuracy.

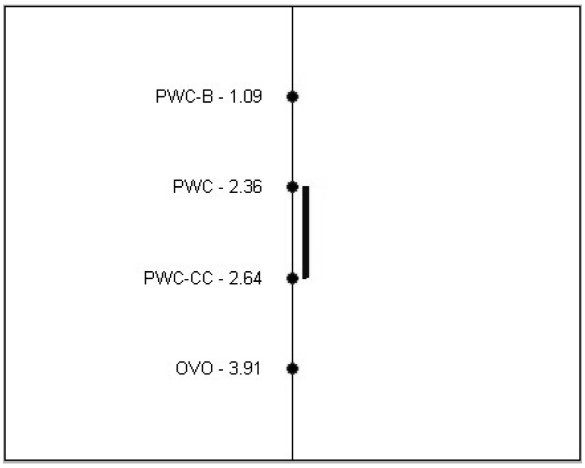

Fig. 9. Comparison of all SVM classifiers against each other with the Nemenyi test. Groups of classifiers that are not significantly different (at $\alpha = 0.1$) are connected.

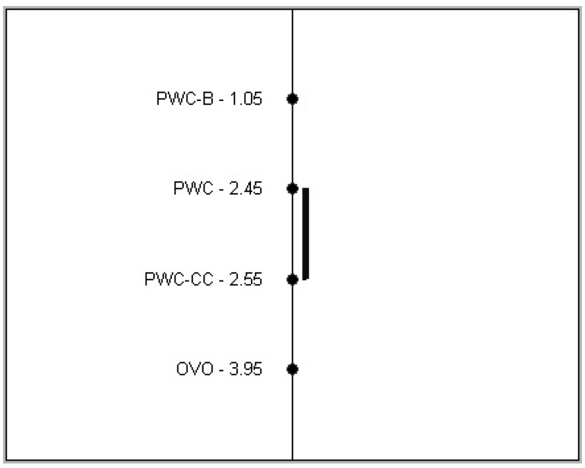

Fig. 10. Comparison of all LDA classifiers against each other with the Nemenyi test. Groups of classifiers that are not significantly different (at $\alpha = 0.1$) are connected.

References

Allwein, E., Schapire, R. and Singer, Y. (2001). Reducing multiclass to binary: A unifying approach for margin classifiers, *Journal of Machine Learning Research* **1**: 113–141.

Beyan, C. and Fisher, R. (2015). Classifying imbalanced data sets using similarity based hierarchical decomposition, *Pattern Recognition* **48**(5): 1653–1672.

Breiman, L. (1996). Bagging predictors, *Machine Learning* **24**(2): 123–140.

Cateni, S., Colla, V. and Vannucci, M. (2014). A method for resampling imbalanced datasets in binary classification tasks for real-world problems, *Neurocomputing* **135**: 32–41.

Chang, C. and Lin, C. (2001). LIBSVM: A library for support vector machines, `http://www.csie.ntu.edu.tw/jlin/libsvm`.

Chawla, N., Bowyer, K., Hall, L. and Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research* **16**: 321–357.

Chmielnicki, W., Roterman-Konieczna, I. and Stąpor, K. (2012). An improved protein fold recognition with support vector machines, *Expert Systems* **20**(2): 200–211.

Chmielnicki, W. and Stąpor, K. (2010). Protein fold recognition with combined SVM-RDA classifier, *in* M.G. Romay and E. Corchado (Eds.), *Hybrid Artificial Intelligence Systems*, Lecture Notes in Artificial Intelligence, Vol. 6076, Springer, Berlin, pp. 162–169.

Chmielnicki, W. and Stąpor, K. (2012). A hybrid discriminative/generative approach to protein fold recognition, *Neurocomputing* **75**(1): 194–198.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research* **7**: 1–30.

Dietterich, T. (1998). Approximate statistical tests for comparing supervised classification learning algorithms, *Neural Computation* **10**: 1895–1924.

Dietterich, T.G. and Bakiri, G. (1995). Solving multiclass problems via error-correcting output codes, *Journal of Artificial Intelligence Research* **2**: 263–286.

Ding, C. and Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks, *Bioinformatics* **17**(4): 349–358.

Fei, B. and Liu, J. (2006). Binary tree of SVM: A new fast multiclass training and classification algorithm, *IEEE Transactions on Neural Networks* **17**(3): 696–704.

Friedman, J. (1996). Another approach to polychotomous classification, *Technical report*, Stanford University, Stanford, CA.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H. and Herrera, F. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, *Pattern Recognition* **44**(8): 1761–1776.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H. and Herrera, F. (2013). Dynamic classifier selection for one-vs-one strategy: Avoiding non-competent classifiers, *Pattern Recognition* **46**(12): 3412–3424.

Glomb, P., Romaszewski, M., Opozda, S. and Sochan, A. (2011). Choosing and modeling hand gesture database for natural user interface, *Proceedings of the 9th International Conference on Gesture and Sign Language in Human-Computer Interaction and Embodied Communication, Athens, Greece*, pp. 24–35.

Hastie, T. and Tibshirani, R. (1998). Classification by pairwise coupling, *The Annals of Statistics* **26**(1): 451–471.

He, H. and Garcia, E. (2009). Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering* **21**(9): 1263–1284.

Hollander, M. and Wolfe, D. (1973). *Nonparametric Statistical Methods*, John Wiley and Sons, New York, NY.

Iman, R. and Davenport, J. (1980). Approximations of the critical region of the Friedman statistics, *Communications in Statistics—Theory and Methods* **9**(6): 571–595.

Kahsay, L., Schwenker, F. and Palm, G. (2005). Comparison of multiclass SVM decomposition schemes for visual object recognition, *in* W. Kropatsch *et al.* (Eds.), *Pattern Recognition*, Lecture Notes in Computer Science, Vol. 3663, Springer, Berlin, pp. 334–341.

Kijsirikul, B. and Ussivakul, N. (2002). Multiclass support vector machines using adaptive directed acyclic graph, *Proceedings of the International Joint Conference on Neural Networks, Honolulu, HI, USA*, pp. 980–985.

Krawczyk, B., Wozniak, M. and Cyganek, B. (2014). Clusterting-based ensembles for one-class classification, *Information Sciences* **264**: 182–195.

Krzysko, M. and Wolynski, W. (2009). New variants of pairwise classification, *European Journal of Operational Research* **199**(2): 512–519.

LeCun, Y., Cortes, C. and Burges, Ch.J.C. (2014). The MNIST database of handwritten digits, `http://yann.lecun.com/ exdb/mnist/`.

Liu, C. and Fujisava, H. (2005). Classification and learning for character recognition: Comparison of methods and remaining problems, *International Workshop on Neural Networks and Learning in Document Analysis and Recognition, Seoul, Korea*, pp. 1–7.

Liu, X., Wu, J. and Zhou, Z.H. (2008). Exploratory undersampling for class-imbalance learning, *IEEE Transactions on Systems, Man and Cybernetics B* **39**(2): 539–550.

Lorena, A. and Carvalho, A. (2010). Building binary-tree-based multiclass classifiers using separability measures, *Neurocomputing* **73**(16–18): 2837–2845.

Lorena, A., Carvalho, A. and Gama, J. (2008). A review on the combination of binary classifiers in multiclass problems, *Artificial Intelligence Review* **30**(1–4): 19–37.

Moreira, M. and Mayoraz, E. (1998). Improved pairwise coupling classification with correcting classifiers, *Proceedings of the 10th European Conference on Machine Learning, ECML 1998, Chemnitz, Germany*, pp. 160–171.

Nadeau, C. and Bengio, Y. (2003). Inference for the generalization error, *Advances in Neural Information Processing Systems* **52**(3): 239–281.

Nemenyi, P. (1963). *Distribution-free Multiple Comparisons*, Ph.D. thesis, Princeton University, Princeton, NJ.

Ou, G. and Murphey, Y. (2006). Multi-class pattern classification using neural networks, *Pattern Recognition* **40**(1): 4–18.

Platt, J., Cristianini, N. and Shawe-Taylor, J. (2000). Large margin DAGs for multiclass classification, *Neural Information Processing Systems, NIPS'99, Breckenridge, CO, USA*, pp. 547–553.

Saez, J.A., Galar, M., Luengo, J. and Herrera, F. (2012). A first study on decomposition strategies with data with class noise using decision trees, *Proceedings of the 7th International Conference on Hybrid Artificial Intelligent Systems, Salamanca, Spain*, Part II, pp. 25–35.

UCIMLR (2014). UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/datasets.html`.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*, Springer, New York, NY.

Vural, V. and Dy, J. (2004). A hierarchical method for multi-class support vector machines, *Proceedings of the 21st International Conference on Machine Learning, St. Louis, MO, USA*, pp. 831–838.

Wilcoxon, F. (1945). Individual comparisons by ranking methods, *Biometrics* **1**(6): 80–83.

Wiesław Chmielnicki received a Ph.D. degree from the Polish Academy of Sciences in 2013. Currently he works at the Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Cracow, Poland. He has several years of industrial experience in software development. His research focuses on statistical pattern recognition, machine learning and bioinformatics. His interests also include gesture recognition methods, especially in real-time systems.

Katarzyna Stąpor received an M.Sc. degree in technical and mathematical sciences (1992, 2002), as well as Ph.D. and D.Sc. (habilitation) degrees and the professorial title in computer science from the Silesian University of Technology, Gliwice, Poland (1994, 2001 and 2006, respectively). Since 1992 she has been working at the Institute of Computer Science, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology. She has published over 80 works, including one monograph and three books. Her current research focuses on statistical pattern recognition, computer vision and bioinformatics.

Received: 2 November 2014
Revised: 11 May 2015
Re-revised: 22 July 2015