Random Resampling in the One-Versus-All Strategy for Handling Multi-class Problems

Christos K. Aridas(✉), Stamatios-Aggelos N. Alexandropoulos, Sotiris B. Kotsiantis, and Michael N. Vrahatis

Computational Intelligence Laboratory, Department of Mathematics, University of Patras, GR-26110 Patras, Greece
char@upatras.gr, {alekst,sotos,vrahatis}@math.upatras.gr

Abstract. One of the most common approaches for handling the multi-class classification problem is to divise the original data set into binary subclasses and to use a set of binary classifiers in order to solve the binarization problem. A new method for solving multi-class classification problems is proposed, by incorporating random resampling techniques in the one-versus-all strategy. Specifically, the division used by the proposed method is based on the one-versus-all binarization technique using random resampling for handling the class-imbalance problem arising due to the one-versus-all binarization. The method has been tested extensively on several multiclass classification problems using Support Vector Machines with four different kernels. Experimental results show that the proposed method exhibits a better performance compared to the simple one-versus-all.

Keywords: Multi-class classification · One-versus-all · Random sampling

1 Introduction

Multi-class, also know as multinomial, classification refers to the problem of classifying patterns into three or more categories, whereas, binary classification is the task of classifying patterns into two distinct categories. Some classification algorithms like Decision Trees, Neural Networks and Bayesian Classifiers naturally handle multi-class problems. On the other hand, some other, like Support Vector Machines (SVMs) [4,16] are restricted to binary problems.

The most common approach for the generalization of binary classification to solve multi-class problems is to decompose the problem into several binary sub-problems [21]. Two of the most well-known approaches are: (a) the one-versus-one (OVO) strategy and (b) the one-versus-all (OVA) strategy [27]. The OVO strategy uses a binary classifier to discriminate piecewise the classes, while the OVA strategy uses a binary classifier to distinguish a single class from the rest classes.

G. Boracchi et al. (Eds.): EANN 2017, CCIS 744, pp. 111–121, 2017.
DOI: 10.1007/978-3-319-65172-9_10

The OVO strategy, for a K class problem, trains $K \cdot (K-1)/2$ classifiers. The most straightforward combination is the majority voting rule where each classifier votes for the predicted class and the one with the largest amount of votes is predicted. On the other hand in OVA, K binary classifiers are trained and the decision is made by applying all binary classifiers to an unseen sample x and by predicting the class label for which the corresponding classifier reports the highest confidence score.

It is known that, the OVA strategy introduces class imbalance [26] during the binary reduction, which may lead classifiers towards the new generated majority class. In this research work a method that handles the problem of class imbalance is presented and its performance is measured in several well-known and widely used benchmark data sets.

The rest of the paper is organized as follows: In Sect. 2 similar works are briefly discussed. In Sect. 3 the proposed method is presented and analysed. In addition, experimental results obtained by using twenty multi-class benchmark data sets are exhibited. The paper ends in Sect. 4 with a synopsis and concluding remarks.

2 Related Work

The multi-class categorization problem [21] is one of the most known problems in Computer Science. Allwein *et al.* [1] have proposed a unifying framework to solve this problem by reducing it to various multiple binary problems. To achieve this, they used a margin-based learning algorithm. Specifically, they unified the most popular approaches: (a) each class is compared against all others, (b) all pairs are compared to each other and (c) codes with error-corresponding properties. In their paper, they have proposed a general method for combining the classifiers generated on the binary problem applying to the most well-known classification learning algorithms such as SVMs, AdaBoost, regression and others [17]. The experimental results with SVMs and AdaBoost have shown that this scheme provides an alternative solution to the mostly used multi-class algorithms.

Zadrozny and Elkan [30] have presented a method that solves the multi-class classification problem through class membership probability estimates using the probability estimates which are produced by binary classifiers. Their experimental results, using boosted naive Bayes, have shown that their method has similar classification accuracy to the loss-based decoding method.

One of the biggest difficulties that we have to deal with is the mapping of the multi-class problem onto a set of simpler binary classification problems, especially when we have to deal with hundreds of classes. Due to the fact that many of the statistical classification models do not have natural multi-class extensions, like SVMs, Rocha and Goldenstein [24] have introduced the correlation and joint probability of base binary learners. They have grouped the binary learners based on their independence and with Bayesian techniques they predict the class of new instances. They have also focused on the reduction of the number of the required learners and how to find new learners that complete the original set.

Despite the progress that has been made recently, the extension of SVMs as multi-class classification solvers is still ongoing. Most of the methods that have been proposed build a multi-class classifier by combing several binary classifiers or considering all classes at once. These kind of methods require to test on large-scale problems, a fact that makes them computationally expensive. Hsu and Lin [15], through their experimental work, have compared OVA, OVO and DAGSVM methods and they have shown that the last two methods are more suitable for practical use. In addition, they have indicated that SVM needs fewer vectors, in the case where all classes are considered at once.

Over the last decade many efforts have been made to construct methods with high classification efficiency for multi-class problems. Fei and Liu [9], have proposed Binary Tress of SVM (BTS), a method which decreases the number of binary classifiers without increasing the total complexity of the problem. The results of their work have shown that BTS maintains comparable accuracy and is much faster to be trained than DAGSVM or ECOC, especially in big problems (with a big number of classes).

Wu *et al.* [29], have proposed two multi-class classification methods for obtaining class probabilities. Both of them are easily implementable and more stable than the voting and Hastie-Tibshirani method [13].

The main goal of so called "binarization strategies" is to divide the original set into two classes, in order to address the multi-class classification problem as well as for each class to train a different binary algorithm. For this scope, two techniques are applied, namely: OVA and OVO. Hence, Galar *et al.* [11], have developed an experimental study on these strategies, to examine the potential of different classifiers, such as SVMs, Decision Trees, Instance Based Learning and Rule Based Systems, as well as the performance and robustness of these techniques, supported by several statistical tests. They have concluded that the best binarization technique highly depends on the base classifier and its confidence estimates.

In the attempt to exploit both the advantages of efficient computations and high classification accuracy, Cheong *et al.* [2], have proposed a binary Decision Tree architecture with SVMs classification. They have introduced a modified version of SOM, the K-SOM, which assists to the achievement of the conversion of multi-class problem into binary tree in order for the decision to be made by SVM. Their method overcomes the performance of trees and maintains comparable classification accuracy in comparison to OVA and OVO strategies.

Lorena *et al.* [21], have presented a survey of the main methods of binary classifiers that can be applied to multi-class classification problems. Another attempt to tackle the multi-class classification problem has been made by Crammer and Singer [5] who have focused on designing output codes and especially continuous codes that have not been viewed as a constrained optimization problem. More specifically, one of the aspects of their formulation was a scheme that built SVMs.

In [12], García-Pedrajas and Ortiz-Boyer have presented many capable binary classifier fusion methods for a multi-class classification problem. These methods

require different assumptions, diverse influences and many aspects that need further study, in order to find out which of those techniques are better for a given multi-class classification problem. Following what we have mentioned above, Duan and Keerthi [8] have conducted an experimental study, trying to conclude on which multi-class SVM method is better.

In the literature, there are various other approaches for the above mentioned issues [7,20,25,28]. Finally, we would like to dwell on the work of Chmielnicki and Stapor [3] who have used instance balancing to improve the performance of pairwise coupling, through the OVA strategy.

3 Proposed Method and Experimental Evaluation

Even in data sets where the patterns are equally distributed between the classes, the OVA approach could lead to high imbalanced binary data sets for each underlined classifier. For example, considering a ten class problem where the patterns are equally distributed, the OVA approach would train binary classifiers that would contain only 10% from the one class and 90% from the other. It is known that many supervised learning algorithms tend to prefer the more common classes using the prior knowledge of the training data set [26]. The proposed approach tackles the problem of class imbalance in data level, independently for each binary classifier, either by random over-sampling (ROS) the minority classes or by random under-sampling (RUS) the majority class. The proposed method is illustrated in Algorithm 1.

For the experiments twenty multi-class data sets have been chosen from the UCI Machine Learning Repository [19]. In Table 1 the name, the number of patterns, the number of input attributes, the number of different classes, as well as the percentage of the majority class for each data set are exhibited. All data sets have been preprocessed following the approach of [10]. Specifically, all discrete input attributes have been transformed to numeric by using a simple quantization. Each attribute has been scaled to have zero mean and standard deviation one. Also, all missing values have been treated as zero.

The classifiers' performance have been measured using the stratified 5-fold cross-validation procedure. The whole data set has been divided into five mutually exclusive folds and for each fold the classifier has been trained on the union of all of the other folds. The folds have been made by preserving the percentage of patterns for each class. Then, cross-validation has been run five times for each algorithm and the mean value of the five folds has been calculated. The performance metric that is reported is the F_1 score which is the weighted average of the precision and recall and has been calculated as:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

All experiments have been conducted with Python using the available implementations from the scikit-learn [23] and imbalanced-learn [18] libraries.

Algorithm 1

parameters
Random Sampling Method M
Base Classifier C
Ratio R
procedure TRAINING(Data Set D)
$f \leftarrow \emptyset$
$m \leftarrow getNumberOfClasses(D)$
for i to m **do**
$onesDataset \leftarrow \emptyset$
$allDataset \leftarrow \emptyset$
for $\forall\ (x, y) \in D$ **do**
if $y <> i$ **then**
$y \leftarrow 0$
$allDataset \leftarrow allDataset \cup \{(x, y)\}$
else
$y \leftarrow 1$
$onesDataset \leftarrow onesDataset \cup \{(x, y)\}$
end if
end for
if $M ==$ 'ros' **then**
$onesDataset \leftarrow randomOverSample(onesDataset, R)$
else
$allDataset \leftarrow randomUnderSample(allDataset, R)$
end if
$binaryDataset \leftarrow onesDataset \cup allDataset$
$f_i \leftarrow trainClassifier(C, binaryDataset)$
end for
end procedure
procedure CLASSIFICATION(Data Set D)
for $\forall\ (x, y) \in D$ **do**
$f(x) \leftarrow \underset{i}{\operatorname{argmax}}\, f_i(x)$
end for
end procedure

The experiments have been carried out using Support Vector Machines with four different kernels. The linear, the polynomial (with degree of 3), the RBF as well as the sigmoid kernel functions have been considered resulting to four different classifiers. The standard OVA approach has been compared to the proposed method using ROS and RUS with base classifier SVMs using the four different kernels mentioned.

In Tables 2, 3, 4 and 5 the obtained results for each kernel are exhibited. The best performer scheme for each data set is reported using boldface digits. It can be easily seen that the proposed method using ROS is the out-performer across all kernel functions, followed by the the RUS version.

The significance of the results have been examined using non-parametric statistical tests [6]. Particularly, the non-parametric Friedman Aligned Ranks [14]

Table 1. Collection of 20 multiclass data sets from the UCI Machine Learning Repository. The number of patterns (#patterns), the number of inputs (#inputs), the number of classes (#classes) as well as the percentage of the majority class (%majority) for each data set, are exhibited.

Data set	#patterns	#inputs	#classes	%majority
abalone	4177	8	3	34.6
arrhythmia	452	262	13	54.2
car	1728	6	4	70.0
contrac	1473	9	3	42.7
dermatology	366	34	6	30.6
ecoli	336	7	8	42.6
flags	194	28	8	30.9
glass	214	9	6	35.5
heart-cleveland	303	13	5	54.1
iris	150	4	3	33.3
lenses	24	4	3	62.5
libras	360	90	15	6.7
low-res-spect	531	100	9	51.9
nursery	12960	8	5	33.3
page-blocks	5473	10	5	89.8
seeds	210	7	3	33.3
steel-plates	1941	27	7	34.7
teaching	151	5	3	34.4
wine-quality-red	1599	11	6	42.6
wine-quality-white	4898	11	7	44.9

test has been performed because of the small number of the compared algorithms. The null-hypothesis states that the performance of all the compared methods are equivalent and therefore their ranks should be equal. In Table 6 the results obtained by using the Friedman test is presented. The p-value in all the cases, with the exception of the polynomial kernel, indicates that the null-hypotheses should be rejected. This means that there are methods whose performance difference was statistically significant to the others. Therefore, post-hoc tests using Nemenyi's [6] procedure have been employed and the obtained results are presented in Table 7. In the case of the linear kernel it can be seen that the ROS version outperforms both versions of the RUS and the standard approach, while in RBF and sigmoid kernels, the standard approach of the OVA scheme has been outperformed by both the ROS and RUS variation.

Table 2. Macro-averaged $F1$ scores using as base classifier SVM with linear kernel.

Data set	Standard-OVA	ROS-OVA	RUS-OVA
abalone	0.485 ± 0.02	$\mathbf{0.626 \pm 0.01}$	0.617 ± 0.02
arrhythmia	$\mathbf{0.422 \pm 0.08}$	0.407 ± 0.10	0.298 ± 0.05
car	0.441 ± 0.09	$\mathbf{0.541 \pm 0.09}$	0.475 ± 0.06
contrac	0.373 ± 0.03	0.490 ± 0.01	$\mathbf{0.487 \pm 0.02}$
dermatology	$\mathbf{0.972 \pm 0.03}$	0.967 ± 0.02	0.969 ± 0.02
ecoli	0.665 ± 0.15	0.609 ± 0.03	$\mathbf{0.668 \pm 0.12}$
flags	$\mathbf{0.294 \pm 0.06}$	0.289 ± 0.07	0.255 ± 0.09
glass	0.340 ± 0.13	$\mathbf{0.483 \pm 0.21}$	0.292 ± 0.11
heart-cleveland	0.283 ± 0.05	$\mathbf{0.383 \pm 0.08}$	0.379 ± 0.07
iris	0.831 ± 0.07	$\mathbf{0.885 \pm 0.04}$	$\mathbf{0.885 \pm 0.07}$
lenses	$\mathbf{0.849 \pm 0.17}$	0.773 ± 0.17	0.698 ± 0.21
libras	$\mathbf{0.568 \pm 0.10}$	0.564 ± 0.11	0.376 ± 0.07
low-res-spect	0.604 ± 0.13	$\mathbf{0.606 \pm 0.06}$	0.500 ± 0.08
nursery	0.518 ± 0.12	$\mathbf{0.537 \pm 0.12}$	0.429 ± 0.11
page-blocks	0.404 ± 0.11	$\mathbf{0.507 \pm 0.13}$	0.469 ± 0.10
seeds	0.909 ± 0.05	$\mathbf{0.928 \pm 0.04}$	0.919 ± 0.04
steel-plates	0.517 ± 0.08	$\mathbf{0.578 \pm 0.08}$	0.546 ± 0.09
teaching	0.405 ± 0.10	$\mathbf{0.529 \pm 0.08}$	0.513 ± 0.09
wine-quality-red	0.190 ± 0.03	0.271 ± 0.03	$\mathbf{0.273 \pm 0.02}$
wine-quality-white	0.175 ± 0.04	$\mathbf{0.262 \pm 0.07}$	0.248 ± 0.03

Table 3. Macro-averaged $F1$ scores using as base classifier SVM with polynomial kernel.

Data set	Standard-OVA	ROS-OVA	RUS-OVA
abalone	0.421 ± 0.03	$\mathbf{0.555 \pm 0.03}$	0.440 ± 0.05
arrhythmia	0.060 ± 0.01	$\mathbf{0.294 \pm 0.06}$	0.282 ± 0.08
car	0.448 ± 0.14	$\mathbf{0.516 \pm 0.12}$	0.495 ± 0.03
contrac	0.363 ± 0.01	$\mathbf{0.468 \pm 0.03}$	0.463 ± 0.02
dermatology	$\mathbf{0.801 \pm 0.09}$	0.616 ± 0.12	0.433 ± 0.12
ecoli	0.205 ± 0.03	$\mathbf{0.446 \pm 0.09}$	0.390 ± 0.09
flags	$\mathbf{0.278 \pm 0.10}$	0.126 ± 0.06	0.115 ± 0.04
glass	0.324 ± 0.07	$\mathbf{0.361 \pm 0.06}$	0.228 ± 0.03
heart-cleveland	0.155 ± 0.02	0.311 ± 0.04	$\mathbf{0.324 \pm 0.05}$
iris	0.532 ± 0.01	0.610 ± 0.14	$\mathbf{0.699 \pm 0.21}$
lenses	0.286 ± 0.08	0.621 ± 0.13	$\mathbf{0.667} \pm 0.24$
libras	$\mathbf{0.598 \pm 0.12}$	0.338 ± 0.15	0.332 ± 0.15
low-res-spect	0.089 ± 0.01	$\mathbf{0.463 \pm 0.07}$	0.203 ± 0.07
nursery	$\mathbf{0.524 \pm 0.11}$	0.516 ± 0.11	0.428 ± 0.09
page-blocks	0.190 ± 0.00	$\mathbf{0.432 \pm 0.05}$	0.308 ± 0.06
seeds	0.540 ± 0.01	0.694 ± 0.16	$\mathbf{0.699 \pm 0.22}$
steel-plates	0.432 ± 0.09	$\mathbf{0.489 \pm 0.08}$	0.429 ± 0.05
teaching	0.468 ± 0.10	$\mathbf{0.488 \pm 0.07}$	0.487 ± 0.09
wine-quality-red	0.173 ± 0.02	0.081 ± 0.06	$\mathbf{0.174 \pm 0.04}$
wine-quality-white	$\mathbf{0.191 \pm 0.03}$	0.092 ± 0.03	0.109 ± 0.03

Table 4. Macro-averaged $F1$ scores using as base classifier SVM with RBF kernel.

Data set	Standard-OVA	ROS-OVA	RUS-OVA
abalone	0.462 ± 0.03	$\mathbf{0.611 \pm 0.03}$	0.603 ± 0.03
arrhythmia	0.060 ± 0.01	$\mathbf{0.361 \pm 0.08}$	0.309 ± 0.04
car	0.492 ± 0.18	$\mathbf{0.637 \pm 0.17}$	0.533 ± 0.06
contrac	0.391 ± 0.03	$\mathbf{0.472 \pm 0.02}$	0.470 ± 0.02
dermatology	0.966 ± 0.02	$\mathbf{0.970 \pm 0.03}$	0.948 ± 0.03
ecoli	0.549 ± 0.13	0.623 ± 0.04	$\mathbf{0.690 \pm 0.07}$
flags	0.258 ± 0.10	0.273 ± 0.05	$\mathbf{0.293 \pm 0.05}$
glass	0.261 ± 0.04	$\mathbf{0.404 \pm 0.17}$	0.262 ± 0.08
heart-cleveland	0.204 ± 0.03	0.345 ± 0.04	$\mathbf{0.349 \pm 0.06}$
iris	0.911 ± 0.08	$\mathbf{0.926 \pm 0.02}$	0.872 ± 0.04
lenses	0.663 ± 0.33	$\mathbf{0.849 \pm 0.17}$	$\mathbf{0.849 \pm 0.17}$
libras	$\mathbf{0.578 \pm 0.10}$	0.460 ± 0.08	0.456 ± 0.10
low-res-spect	0.373 ± 0.09	$\mathbf{0.454 \pm 0.06}$	0.343 ± 0.05
nursery	0.533 ± 0.10	$\mathbf{0.568 \pm 0.10}$	0.457 ± 0.08
page-blocks	0.311 ± 0.08	$\mathbf{0.506 \pm 0.11}$	0.493 ± 0.11
seeds	0.914 ± 0.06	0.914 ± 0.05	$\mathbf{0.918 \pm 0.05}$
steel-plates	0.484 ± 0.05	$\mathbf{0.582 \pm 0.08}$	0.519 ± 0.07
teaching	0.455 ± 0.11	$\mathbf{0.505 \pm 0.06}$	0.486 ± 0.09
wine-quality-red	0.177 ± 0.04	$\mathbf{0.266 \pm 0.02}$	0.259 ± 0.03
wine-quality-white	0.181 ± 0.04	$\mathbf{0.271 \pm 0.07}$	0.224 ± 0.01

Table 5. Macro-averaged $F1$ scores using as base classifier SVM with sigmoid kernel.

Data set	Standard-OVA	ROS-OVA	RUS-OVA
abalone	0.453 ± 0.03	$\mathbf{0.583 \pm 0.03}$	0.572 ± 0.04
arrhythmia	0.006 ± 0.01	$\mathbf{0.368 \pm 0.08}$	0.304 ± 0.04
car	0.298 ± 0.01	$\mathbf{0.411 \pm 0.06}$	0.389 ± 0.06
contrac	0.393 ± 0.02	$\mathbf{0.466 \pm 0.02}$	0.459 ± 0.02
dermatology	0.957 ± 0.02	$\mathbf{0.967 \pm 0.01}$	0.935 ± 0.04
ecoli	0.493 ± 0.12	0.590 ± 0.06	$\mathbf{0.611 \pm 0.14}$
flags	0.248 ± 0.07	0.263 ± 0.05	$\mathbf{0.269 \pm 0.08}$
glass	0.248 ± 0.05	$\mathbf{0.348 \pm 0.17}$	0.244 ± 0.04
heart-cleveland	0.231 ± 0.04	$\mathbf{0.392 \pm 0.07}$	0.289 ± 0.03
iris	0.573 ± 0.06	0.626 ± 0.08	$\mathbf{0.778 \pm 0.05}$
lenses	0.286 ± 0.08	0.760 ± 0.18	$\mathbf{0.849 \pm 0.17}$
libras	$\mathbf{0.441 \pm 0.09}$	0.405 ± 0.13	0.367 ± 0.15
low-res-spect	0.281 ± 0.10	$\mathbf{0.448 \pm 0.08}$	0.265 ± 0.06
nursery	0.402 ± 0.07	$\mathbf{0.465 \pm 0.10}$	0.409 ± 0.10
page-blocks	0.260 ± 0.06	0.469 ± 0.11	$\mathbf{0.481 \pm 0.09}$
seeds	0.790 ± 0.03	0.898 ± 0.05	$\mathbf{0.903 \pm 0.04}$
steel-plates	0.451 ± 0.04	$\mathbf{0.573 \pm 0.09}$	0.491 ± 0.05
teaching	$\mathbf{0.503 \pm 0.07}$	0.488 ± 0.13	0.378 ± 0.03
wine-quality-red	0.154 ± 0.02	$\mathbf{0.257 \pm 0.04}$	0.243 ± 0.02
wine-quality-white	0.164 ± 0.02	$\mathbf{0.245 \pm 0.04}$	0.211 ± 0.02

Table 6. Rankings of the algorithms using the Friedman Aligned Ranks test.

SVM-linear		SVM-poly		SVM-RBF		SVM-sigmoid	
ROS	42.42500	ROS	37.85000	ROS	42.95000	ROS	43.25000
RUS	25.32500	RUS	29.95000	RUS	31.82500	RUS	33.05000
Standard	23.75000	Standard	23.70000	Standard	16.72500	Standard	15.20000
Statistic	9.92085	Statistic	4.56181	Statistic	15.82051	Statistic	18.58499
p-value	0.00701	p-value	0.10219	p-value	0.00037	p-value	0.00009

Table 7. Post hoc comparisons using the Nemenyi's procedure.

Comparison	Statistic	Adjusted p-value	Result
SVM-linear			
ROS vs RUS	3.09632	0.00588	H_0 is rejected
ROS vs standard	3.38151	0.00216	H_0 is rejected
Standard vs RUS	0.28519	1.00000	H_0 is not rejected
SVM-RBF			
ROS vs RUS	2.01442	0.13190	H_0 is not rejected
ROS vs standard	4.74860	0.00001	H_0 is rejected
Standard vs RUS	2.73418	0.01876	H_0 is rejected
SVM-sigmoid			
ROS vs RUS	1.84693	0.19427	H_0 is not rejected
ROS vs standard	5.07906	0.00000	H_0 is rejected
Standard vs RUS	3.23213	0.00369	H_0 is rejected

4 Conclusions

An alternative scheme to the one-versus-all strategy for extending binary classifiers to multi-class cases is presented. The proposed scheme tackles the imbalanced problem that is introduced when the one-versus-all strategy decomposes a multi-class problem to several binary ones. Therefore, before training each binary classifier, the imbalanced problem is solved by the usage of a random resampling strategy. Experiments on several standard well-known and widely used benchmark data sets show that the application either of RUS or ROS to each binary classifier could enhance the performance compared to the standard one-versus-all approach. Exploiting the adaptation of the proposed approach in multi-label classification tasks [22], could be an interesting area for further research.

Acknowledgements. Stamatios-Aggelos N. Alexandropoulos gratefully acknowledges the support of his work by the Hellenic State Scholarships Foundation (IKY), co-financed by the European Union (European Social Fund–ESF) and Greek national

funds, “Reinforcement of the Human Research Potential through Doctoral Research” of the Operational Program “Development of Human Capital, Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF 2014–2020).

References

1. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. J. Mach. Learn. Res. **1**, 113–141 (2000)
2. Cheong, S., Oh, S.H., Lee, S.Y.: Support vector machines with binary tree architecture for multi-class classification. Neural Inf. Process. Lett. Rev. **2**(3), 47–51 (2004)
3. Chmielnicki, W., Stąpor, K.: Using the one-versus-rest strategy with samples balancing to improve pairwise coupling classification. Int. J. Appl. Math. Comput. Sci. **26**(1), 191–201 (2016)
4. Christianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
5. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. Mach. Learn. **47**(2), 201–233 (2002)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
7. Dogan, U., Glasmachers, T., Igel, C.: A unified view on multi-class support vector classification. J. Mach. Learn. Res. **17**(45), 1–32 (2016)
8. Duan, K.-B., Keerthi, S.S.: Which is the best multiclass SVM method? an empirical study. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 278–285. Springer, Heidelberg (2005). doi:10.1007/11494683_28
9. Fei, B., Liu, J.: Binary tree of SVM: a new fast multiclass training and classification algorithm. IEEE Trans. Netw. **17**(3), 696–704 (2006)
10. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. **15**, 3133–3181 (2014). bibtex: fernandez-delgado_we_2014. http://jmlr.org/papers/v15/delgado14a.html
11. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes. Pattern Recogn. **44**(8), 1761–1776 (2011)
12. García-Pedrajas, N., Ortiz-Boyer, D.: An empirical study of binary classifier fusion methods for multiclass classification. Inf. Fusion **12**(2), 111–130 (2011)
13. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. Ann. Stat. **26**(2), 451–471 (1998). http://dx.doi.org/10.1214/aos/1028144844
14. Hodges, J.L., Lehmann, E.L.: Rank methods for combination of independent experiments in analysis of variance. In: Rojo, J. (ed.) Selected Works of E.L. Lehmann, pp. 403–418. Springer, Heidelberg (2011). doi:10.1007/978-1-4614-1412-4_35
15. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. **13**(2), 415–425 (2002)
16. Jian, L., Gao, C.: Binary coding SVMs for the multiclass problem of blast furnace system. IEEE Trans. Ind. Electro. **60**(9), 3846–3856 (2013)
17. Kotsiantis, S.B.: Bagging and boosting variants for handling classifications problems: a survey. Knowl. Eng. Rev. **29**(01), 78–100 (2014)

18. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. **18**(17), 1–5 (2017). http://jmlr.org/papers/v18/16-365.html
19. Lichman, M.: UCI Machine Learning Repository (2013). http://archive.ics.uci.edu/ml
20. Liu, M., Zhang, D., Chen, S., Xue, H.: Joint binary classifier learning for ecoc-based multi-class classification. IEEE Trans. Pattern Anal. Mach. Intell. **38**(11), 2335–2341 (2016)
21. Lorena, A.C., De Carvalho, A.C., Gama, J.M.: A review on the combination of binary classifiers in multiclass problems. Artif. Intell. Rev. **30**(1), 19–37 (2008)
22. Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning. Pattern Recogn. **45**(9), 3084–3104 (2012)
23. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
24. Rocha, A., Goldenstein, S.K.: Multiclass from binary: expanding one-versus-all, one-versus-one and ecoc-based approaches. IEEE Trans. Neural Netw. Learn. Syst. **25**(2), 289–302 (2014)
25. Santhanam, V., Morariu, V.I., Harwood, D., Davis, L.S.: A non-parametric approach to extending generic binary classifiers for multi-classification. Pattern Recogn. **58**, 149–158 (2016)
26. Kotsiantis, S., Kanellopoulos, D., Pintelas, P.: Handling imbalanced datasets: a review. Int. Trans. Comput. Sci. Eng. **30**, 25–36 (2006)
27. Tax, D.M., Duin, R.P.: Using two-class classifiers for multiclass classification. In: Proceedings of the 16th IEEE International Conference on Pattern Recognition, vol. 2, pp. 124–127. IEEE (2002)
28. Windeatt, T., Ghaderi, R.: Coding and decoding strategies for multi-class learning problems. Inf. Fusion **4**(1), 11–21 (2003)
29. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. **5**, 975–1005 (2004)
30. Zadrozny, B., Elkan, C.: Reducing multiclass to binary by coupling probability estimates. In: Advances in Neural Information Processing Systems, vol. 2, pp. 1041–1048 (2002)