Information Fusion 36 (2017) 26–51

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

NMC: nearest matrix classification – A new combination model for pruning One-vs-One ensembles by transforming the aggregation problem

Mikel Galar[a,e,*], Alberto Fernández[b], Edurne Barrenechea[a,e], Humberto Bustince[a,e], Francisco Herrera[c,d]

[a] Departamento de Automática y Computación, Universidad Pública de Navarra, 31006, Pamplona, Spain
[b] Department of Computer Science, University of Jaén, 23071, Jaén, Spain
[c] Department of Computer Science and Artificial Intelligence, University of Granada, 18071, Granada, Spain
[d] Faculty of Computing and Information Technology - North Jeddah, King Abdulaziz University, 21589, Jeddah, Saudi Arabia
[e] Institute of Smart Cities (ISC), Universidad Pública de Navarra, 31006, Pamplona, Spain

ARTICLE INFO

Article history:
Received 22 July 2015
Revised 15 February 2016
Accepted 3 November 2016
Available online 4 November 2016

Keywords:
Multi-class classification
One-vs-One
Decomposition strategies
Classifier selection
Ensemble pruning

ABSTRACT

The One-vs-One strategy is among the most used techniques to deal with multi-class problems in Machine Learning. This way, any binary classifier can be used to address the original problem, since one classifier is learned for each possible pair of classes. As in every ensemble method, classifier combination becomes a vital step in the classification process. Even though many combination models have been developed in the literature, none of them have dealt with the possibility of reducing the number of generated classifiers after the training phase, i.e., ensemble pruning, since every classifier is supposed to be necessary.

On this account, our objective in this paper is two-fold: (1) We propose a transformation of the aggregation step, which lead us to a new combination strategy where instances are classified on the basis of the similarities among score-matrices. (2) This fact allows us to introduce the possibility of reducing the number of binary classifiers without affecting the final accuracy. We will show that around 50% of classifiers can be removed (depending on the base learner and the specific problem) and that the confidence degrees obtained by these base classifiers have a strong influence on the improvement in the final accuracy.

A thorough experimental study is carried out in order to show the behavior of the proposed approach in comparison with the state-of-the-art combination models in the One-vs-One strategy. Different classifiers from various Machine Learning paradigms are considered as base classifiers and the results obtained are contrasted with the proper statistical analysis.

1. Introduction

Multi-class problems are present in many real-world applications, for example, the severity grading of diseases [1], fingerprint classification [2], the classification of micro-arrays [3] or people tracking [4] to name a few. Although the number of problems that can be viewed as multi-class ones is increasing, binary classifiers are much more studied in the literature. This is due to the fact that there are some classifier learning paradigms in which multi-class modeling is not straightforward. A well-known example of this situation is Support Vector Machine (SVM) [5].

One simple, yet effective way to address multi-class problems in these cases is by means of decomposition strategies [6]. In order to do so, multi-class problems are divided into easier-to-solve binary classification problems following the divide-and-conquer paradigm. As a result, a set of classifiers is learned, each one being responsible for a binary problem. In the testing phase, the outputs of all the classifiers for a given instance are aggregated to make the final decision [7]. Therefore, the difficulty in addressing the multi-class problem is shifted from the classifier itself to the combination stage.

Among decomposition strategies, the One-vs-One (OVO) [8] scheme stands out as one of the most popular techniques.

* Corresponding author. Fax: +34 948 168924.
E-mail addresses: mikel.galar@unavarra.es (M. Galar), alberto.fernandez@ujaen.es (A. Fernández), edurne.barrenechea@unavarra.es (E. Barrenechea), bustince@unavarra.es (H. Bustince), herrera@decsai.ugr.es (F. Herrera).

http://dx.doi.org/10.1016/j.inffus.2016.11.004
1566-2535/

Its usage to model multi-class problems with SVMs in very well-known software tools such as WEKA [9], LIBSVM [10] or KEEL [11], has made it prevalent in many applications. However, it should be mentioned that this strategy can be included in the broader framework of Error Correcting Output Codes (ECOC) [12,13] In OVO, the multi-class problem division is carried out in such a way that a new binary problem is generated for each possible pair of classes. This is why it is also known as *pairwise learning* [14]. Nevertheless, OVO is not only useful to deal with multi-class problems using classifiers without inherent multi-class support, but it also provides a better classification accuracy than addressing the problem directly using multi-class classifiers [15–19].

In the combination phase, the way in which the problem is divided has to be taken into account as a key factor. Several combination methods for the OVO strategy can be found in the literature [18], among which a voting strategy is the most intuitive one (each classifier votes for its predicted class and the most voted one is given as output). Nonetheless, more elaborated approaches have also been developed attending at the inherent difficulties in the OVO decomposition [20–22], although the same accuracy is achieved by simpler alternatives such as the Weighted Voting (WV) [14] or probability estimation methods [23]. An exhaustive empirical study on the combination methods for OVO can be found in [18], where the presence of non-competent classifiers in this strategy was stressed as a promising research line to improve previous combination models. Non-competent classifiers are those that have not been trained with instances from the class to which the example to be classified belongs to. Recent developments have shown that an effective handling of these classifiers allows one to improve the final classification accuracy rate [24,25].

In this paper our aim is to look at the aggregation phase from a different perspective, which may also take advantage of non-competent classifiers rather than avoiding them. Specifically, in our contribution we transform this aggregation by thinking of the outputs of the classifiers as new inputs to another classification problem, which is used to determine the final class labels of the dataset. This view is similar to Stacking [26], although neither a cross-validation procedure is followed (the same base classifier is used for all subproblems) nor a classifier is trained. Stacking and OVO together have been previously considered but with different purposes to ours, focusing on Stacking with cross-validation using different base classifiers [27] and making use of OVO as a Stacking method [28]. In our case, the main difference appears at the combination method. Once the outputs for every training instance are obtained (each one stored in a score-matrix), new instances are simply classified by the most similar score-matrices to that obtained for the new instance, that is, the k Nearest Neighbors (kNN) [29] classifier is applied over the score-matrices (neither requiring a cross-validation nor the usage of different types of base classifiers). This is why we named it as *Nearest Matrix Classification* (NMC).

We will show that by itself this strategy can be competitive and even superior to the state-of-the-art aggregations, although its behavior strongly depends on the underlying classifier and the quality of its confidence degrees. This fact together with the added storage necessity lead us to introduce prototype (in this case, score-matrix) selection methods [30]. This way, only those score-matrices coming from examples that are useful for the classification are maintained in the reference set for NMC classifier, reducing the storage necessity and improving the classification performance as a result of being more robust with respect to the different base classifiers.

More interestingly, this novel view allows us to introduce pruning techniques [31] into OVO, which have not been previously considered, since all classifiers are supposed to be necessary. Pruning techniques for ensembles aim at reducing the pool of classifiers, decreasing the storage necessity, improving performance and reducing testing times. Our new perspective on the combination phase turns the pruning (i.e., classifier selection) into a feature selection problem [32] for the kNN classifier. We will show that almost half of the classifiers in OVO can be safely removed for testing time (depending on the problem and the base classifier) and that if the appropriate confidence estimates are given by the underlying classifier, accuracy can also be boosted in some cases. In order to carry out the feature and instance selection, we consider the usage of a Genetic Algorithm (GA), which has been previously applied with success [33–35].

All these aspects are analyzed in a thorough experimental study, where twenty three real-world problems from the KEEL data-set repository[1] [11,36] are tested using several well-known classifiers from different Machine Learning paradigms as base learners, namely, SVMs [5], decision trees [37,38], instance-based learning [29], and decision lists [39]. Different evaluation criteria are considered to measure the performance, storage reduction and training times. The conclusions obtained are supported by the appropriate statistical tests as suggested in the literature [40,41]. In addition to NMC classifier, state-of-the-art combinations for OVO [18], including a novel Dynamic Classifier Selection (DCS) approach [24] are included in the empirical comparison.

The contributions of this paper are:

- A new combination strategy for OVO is proposed by transforming the aggregation problem.
- The possibility of carrying out pruning in OVO ensembles is introduced for the first time.
- An exhaustive experimental study showing the existence of redundant (non-necessary) classifiers in OVO is developed, which opens up new future research lines in the topic.

The rest of this paper is organized as follows. Section 2 recalls several concepts used in this work. Afterwards, Section 3 discusses other works related to our proposal. Next, Section 4 presents our NMC proposal to prune OVO ensembles. The set-up of the experimental framework is presented in Section 5, whereas the experimental analysis is carried out in Section 6. Finally, Section 7 concludes the paper and presents the future research lines.

2. Preliminaries

This section recalls the OVO scheme, including existing combinations. Afterwards, DTs and their application in OVO are explained.

2.1. The One-vs-One scheme

In the OVO strategy, a m-class problem is divided into $m(m-1)/2$ two-class problems (one for each possible pair of classes). Each binary classification sub-problem is addressed by a different classifier, which is built using training instances only from the two classes considered. This fact is what causes the *non-competence* problem [14,18,24,25] in testing phase.

An easy way of organizing the outputs of the base classifiers for an instance is by means of a score-matrix R, from which different combination models can be applied:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \quad (1)$$

[1] http://www.keel.es/dataset.php.

where $r_{ij} \in [0, 1]$ represents the confidence of the classifier discriminating classes i and j in favor of the former; whereas the confidence for the latter is computed by $r_{ji} = 1 - r_{ij}$ (if the classifier does not provide it[2]).

2.2. Combination strategies for the One-vs-One scheme

Several strategies for combining the OVO classifiers have been proposed in the literature aiming at achieving the highest accuracy addressing different features of this inference step. In [18], we developed a thorough review and a experimental comparison considering the most-recent and well-known techniques. From this study, we were able to select the better suited combination strategies for different paradigms of classifiers, which are presented hereafter.

- *Weighted Voting strategy* (WV) [14] uses the confidence of each base classifier in each class to vote for it. The class with the largest total confidence is the final output class:

$$Class = arg \max_{i=1,\dots,m} \sum_{1 \le j \ne i \le m} r_{ij} \tag{2}$$

- *Non-Dominance Criterion* (ND) [42] considers the score-matrix as a fuzzy preference relation, which must be normalized. This method predicts the class with the largest degree of non-dominance, that is, the class which is less dominated by all the remaining classes:

$$Class = arg \max_{i=1,\dots,m} \left\{ 1 - \max_{1 \le j \ne i \le m} r'_{ji} \right\} \tag{3}$$

where r'_{ji} corresponds to the normalized and strict score-matrix.
- *Wu*et al.*Probability Estimates by Pairwise Coupling approach* (PE) [23] aims to estimate the posterior probabilities $\mathbf{p} = (p_1, \dots, p_m)$ of all the classes starting from the pairwise class probabilities. Finally, the class having the largest posterior probability is predicted:

$$Class = arg \max_{i=1,\dots,m} p_i \tag{4}$$

The posterior probabilities ($\mathbf{p}$) are computed solving the following optimization problem:

$$\min_{\mathbf{p}} \sum_{i=1}^{m} \sum_{1 \le j \ne i \le m} (r_{ji} p_i - r_{ij} p_j)^2 \quad \text{subject to} \quad \sum_{i=1}^{m} p_i = 1, p_i \ge 0, \text{for all } i \in \{1, \dots, m\}. \tag{5}$$

A more extensive and detailed description of these methods is available in [43].

In addition to these methods, we will consider a novel approach, in which we also aimed at getting rid of the non-competent classifiers by means of a DCS strategy [24]. In that work, only the classifiers whose classes were in the neighborhood of the instance were considered. The size of the neighborhood used was large ($3 \cdot m$) compared with the usually considered one for kNN [30]. This work is related to our model since a classifier selection is carried out, but in NMC this selection is static (what is known as pruning), and hence equal to all the instances, whereas in DCS it is dependent on each instance.

Remark 1. All these methods use exactly the same score-matrix values (Eq. (1)) to compute the final class, but they can obtain different results. We must emphasize the importance of this fact, since it allows us to fix the score-matrices of each base classifier, applying the combinations to the same outputs; hence, all the results shown in the experimental analysis will be due to the combinations themselves and not due to differences on the predictions of the base classifiers.

[2] If the classifier provides both confidence degrees, one must ensure that they are normalized such that $r_{ij} + r_{ji} = 1$.

2.3. Decision templates for the One-vs-One decomposition

Decision Templates (DTs) [44] are a well-known fusion strategy in classifier ensembles, which are related to our proposal, and this is why we also consider them in the experimental study. They are based on computing the average output of each classifier for the examples of each class, storing them in different templates. Afterwards, new examples are classified by finding the most similar template to the outputs given by the classifiers for the instance.

This philosophy can be translated to decomposition models even though they were designed to work with classical ensembles. In OVO, DTs can be computed as the average score-matrices for all the instances of each class (only the upper or lower triangular matrix is needed, since both are complementary). That is, score-matrices (R) act as what were originally named as Decision Profiles (DPs) whose averaging per class form the DTs.

In order to classify a new instance, its DP is obtained by submitting it to all classifiers. This DP is then compared with all the DTs using a similarity measure (distance from the template), taking the predicted class from the DT whose similarity is the highest:

$$Class = arg \max_{i=1,\dots,m} \mathcal{J}(DT_i, DP(x)) \tag{6}$$

where x is the instance to be classified, DT_i is the DT of the ith class, $DP(x)$ is the DP obtained for instance x and J is any similarity measure. In this paper, we have considered the most popular similarity measure, which is based on the Euclidean distance:

$$\mathcal{J}(DT_i, DP(x)) = 1 - \frac{1}{L} \sum_{j=1}^{L} (DT_i(j) - d_j(x))^2 \tag{7}$$

where L is the number of classifiers ($m(m-1)/2$ in OVO), $DT_i(j)$ is the jth element of the ith DT (corresponding to classifier j) and $d_j(x)$ is the output of the jth classifier for instance x. We will show that this representation of the outputs can also be used to prune OVO ensembles.

3. Related work

Ensemble pruning techniques are designed to reduce the storage necessity, testing times and even increase the accuracy of classifier ensembles [31,45,46]. These methods assume the fact that every classifier is able to distinguish all the classes of the problem, which in decomposition strategies does not occur. In fact, whereas in classical ensembles the base classifiers usually vary in terms of the input space, in decomposition strategies their differences appear at the output space. This is why these types of ensemble pruning techniques are not directly applicable to these strategies. Moreover, in the OVO approach all the classifiers are assumed to be needed, since each classifier is responsible for a different pair of classes.

In other respect, the non-competent classifier handling in OVO has been shown to be important in order to improve the accuracy of this model, showing that their presence could harm some of the predictions [24,25]. A DCS model was developed in [24], where only those classifiers that were most probably competent for the instance to be classified were used. In [25], classifiers were not completely removed, but weighted depending on the distance between the instance and each class. With a similar idea, reliability maps were proposed in the wider framework of ECOC [47]. Otherwise, García-Pedrajas [48] combined the OVA and OVO strategies avoiding the non-competence problem. A similar combination was also developed in [49].

Other approaches in OVO have dealt with the problem of reducing the number of classifier tested when an instance is to be classified. This are the cases of DDAG [22] and BTC [20]. Another efficient model to test classifiers in ECOC framework was developed in [50] (and hence, applicable to OVO), where exactly the same accuracy as that obtained with the original model was achieved, but with a great reduction in the number of classifiers tested. Different to our proposal, these models did not considered the removal of classifiers before the testing phase.

Also dealing with the efficiency of ECOC ensembles, Rocha and Goldestein [51] developed a method where starting from a random ECOC, the classifiers were added and removed from the ensemble. Focused on data-sets with large number of classes they were able to achieve a low decrease in accuracy while maintaining the efficiency due to the low number of classifiers considered. However, in most of the data-sets where OVO was applied, OVO itself was capable of achieving the best performance in spite of its simplicity. Moreover, the authors showed how OVO lost precision as classifiers were randomly removed from it. Nevertheless, we will show that this was due to the combination used, since OVO allows for removing some of the classifiers without loss of accuracy, if the combination is properly designed. Other similar techniques mainly focused on reducing the number of classifiers needed in problems with a large number of classes are those focused on hierarchical approaches [52,53], in which the reduction is achieved by an increase in building complexity (of the hierarchy and the classifiers themselves).

Stacking [26] can also be related to our approach as mentioned in the introduction. In this technique a meta-classifier is learned on top of the outputs of the classifiers in a multiple classifier system (MCS). Our aim differs from stacking in the sense that it aims to correct and learn from the errors of the classifiers, whereas we assume that instances of the same classes should behave similarly, giving similar outputs for the same base classifiers (even if they are not competent). In [28], Stacking was not used to combine the OVO classifiers, but the OVO strategy was used in a multi-layer Stacking procedure, and more specifically, in its first layer in order to improve the classical Stacking model for multi-class problems. Finally, the approach considered in [27] was different, the authors considered an OVO model where base classifiers coming from different machine learning paradigms were considered for each subproblem. Then, Stacking was considered to combine such a great number of classifiers ($m(m-1)/2 \cdot L$, where L is the number of different base classifiers considered). Their aim was to find the best combination of base classifiers for each pair of classes, and even though a selection procedure was established, few classifiers were removed and many more than in standard OVO were finally considered. In these approaches focused on improving accuracy considering Stacking, different base classifiers are required in order to apply the cross-validation procedure. In contrast, our method focuses on pruning classifiers from the standard OVO models, where the same base classifier is used to classify every pair of classes.

4. NMC: nearest matrix classification

In this section, we present our new combination proposal for the OVO strategy. First, we introduce the basic idea of the method and the hypothesis that has motivated our approach (Section 4.1). Afterwards, we present three possible ways in which it can be extended to select the appropriate instances (Section 4.2), prune classifiers (Section 4.3) or doing both tasks simultaneously (Section 4.4). Finally, we describe the details of the optimization procedure followed to carry out the proposed reduction approach (Section 4.5).

4.1. Basic idea and hypothesis

Our idea comes from the basic assumption that instances belonging to the same class should obtain similar score-matrices after being submitted to all the classifiers. In fact, we assume that this should also occur even in the case that classifiers are non-competent for the instances.

With this basic hypothesis one can leave classical aggregations aside and focus on predicting the class based on the score-matrices of the training examples. Hence, once the score-matrix for a new instance is obtained, the k-NN of the matrix (the k most similar matrices) are computed. The predicted class is obtained from the most repeated class among these neighbors (whose associated class is the one of the original instance, i.e., the classical aggregation step is no longer applied).

Notice that in this view of the score-matrix, it contains redundant information, since each element r_{ij} can be computed from r_{ji} as $r_{ij} = 1 - r_{ji}$. For this reason, we only consider the upper triangular matrix (also reducing the dimensionality of the problem). For an easier understanding of the problem, each score-matrix is transformed to a vector of length $m(m-1)/2$ as follows:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \Longrightarrow \hat{R}$$
$$= (r_{12}, \ldots, r_{1m}, r_{23}, \ldots, r_{2m}, \ldots, r_{(m-1)m}) \quad (8)$$

Therefore, for each training instance $\mathbf{x}_i$, whose class label is y_i (for $i = 1, \ldots, n$, being n the number of examples in the training set), the corresponding score-matrix $R(\mathbf{x}_i)$ is transformed to a vector $\hat{R}(\mathbf{x}_i)$, which becomes a new instance with class y_i for the reference set of NMC. A new instance $\mathbf{x}$ whose score-matrix $R(\mathbf{x})$ is transformed to a vector $\hat{R}(\mathbf{x})$ is classified by the predominant class in its k closest $\hat{R}(\mathbf{x}_i)$ vectors. As we are dealing with numerical values, we consider the Euclidean distance to compute them:

$$d(\hat{R}(\mathbf{x}_i), \hat{R}(\mathbf{x})) = \sqrt{\sum_{j=1}^{L} (\hat{R}_j(\mathbf{x}_i) - \hat{R}_j(\mathbf{x}))^2} \quad (9)$$

where L is the length of each vector (number of classifiers) and $\hat{R}_j(\mathbf{x})$ is the jth element of vector $\hat{R}$. The main difference with respect to DTs is that in this case we have as many templates as examples instead of having a unique template for each class.

It is clear that this fact adds complexity to the aggregation phase, since it involves the computations of the kNNs. In order to alleviate this negative effect, we propose to carry out an instance selection (IS) mechanism, also known as prototype selection (explained in Section 4.2). In particular, for this purpose we have selected an hybrid method to reduce the reference set in order to make kNN faster (condensing), but also to remove instances hindering the kNN classification (editing) based on genetic algorithms (GAs) [30].

Furthermore, not only can we perform an IS, but this new representation also allows us to carry out a feature selection (FS) on the reference set [32]. In fact, this is how we are able to prune OVO ensembles, since removing features from this set is equivalent to removing classifiers. This FS also referred to as classifier selection (CS) is explained in Section 4.3.

It may become evident that we can also perform both tasks at the same time, which is explained in Section 4.4. In the three cases, we have opted for a GA named as CHC (Crossover elitism population, Half uniform crossover combination, Cataclysm mutation) [54] due to its excellent behavior in this context [30,33,35]. It is one of the best performing models for these purposes and it

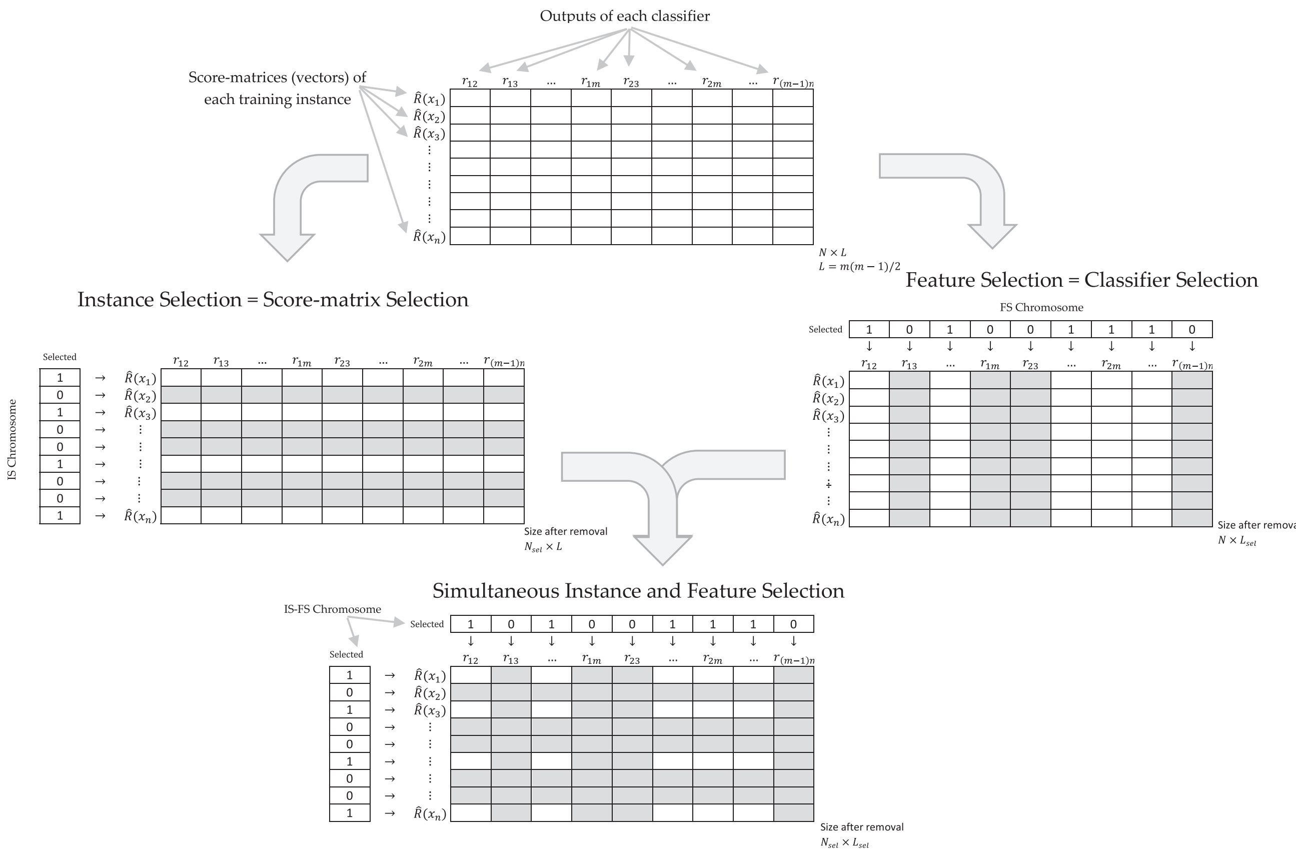

Fig. 1. Schematic representation of the three reduction methods (IS, CS and IS_CS).

can scale to hundreds of thousands of instances with the existing implementations for large-scale data-sets [34].

As a summary, one can think of a matrix where each vector $\hat{R}$ is a row and each column refers to a classifier. IS removes rows from the matrix, whereas CS removes columns. Likewise, doing both tasks at the same time, rows and columns are simultaneously removed, highly reducing the testing times and storage necessities. With this view, IS and CS using GAs can be easily explained as shown in Fig. 1 (where chromosomes encode whether instances or features are considered as explained in next sections).

4.2. Instance selection or score-matrix selection

Instance selection (IS), also known as prototype selection, is a well-known technique mainly used for the kNN classifier (we refer the reader to [30] for an exhaustive review, including an experimental study). The aim of these methods is two-fold: (1) to reduce the reference set; (2) to improve accuracy. In our case, we look for performing both objectives simultaneously, and hence we focus on hybrid methods.

Among hybrid IS methods, we have opted for the one based on a GA called CHC, since it offers a great compromise between accuracy and reduction (as concluded in the experimental study by García et al. [30]). Moreover, it has been successfully applied in different scenarios for IS [35,55,56] and there are implementations of this algorithm that scale up to millions of instances [34], even though we focus on its original implementation, which is enough for the purpose of this work.

The idea behind this algorithm consists of finding the minimum subset of instances achieving the maximum accuracy. In order to do so, chromosomes in the GA encode which instances are selected for the reference set (left side of the scheme in Fig. 1). Hence, a binary chromosome represents a solution:

$$C_{IS} = (c_{x_1}, c_{x_2}, c_{x_3}, c_{x_4}, \ldots, c_{x_N}), \tag{10}$$

where $c_{x_i} \in \{0, 1\}$, indicating whether instance x_i is included or not in the reference set (N being the number of instances).

As in every GA, the population formed of chromosomes is evolved based on the quality of each member, which is measured by the so-called fitness function (f). The main objective of the optimization process is to maximize the predictive accuracy of the ensemble (i.e., the percentage of correctly classified instances). However, the objective of reducing instances from the training set can also be taken into account in the fitness function. For this reason, we study the effect of different fitness functions in the experimental study to analyze how forcing reduction affects accuracy. These functions are described hereafter:

1. **Acc** (f_A): The instance subset maximizing accuracy is sought.
2. **AccRedIS** (f_{AI}): A lexicographical order is established. Accuracy is taken into account first. If there is a tie, the chromosome with the greatest instance reduction is selected.

Whereas reduction can be directly computed from the chromosome (as the percentage of 0's), accuracy needs to be computed by kNN using the instances selected as reference set . In order to do so, as usual, accuracy is computed by leave-one-out. Details about CHC are given in Section 4.5, since the three models make use of it. Notice that in all cases, if ties still continue, they are solved in favor of the parents or randomly otherwise.

4.3. Feature selection or classifier selection (pruning)

In this case, the idea is to remove those classifiers which do not contribute to the correct classification or that can be redundant.

As in the case of IS, there are a number of FS methods in the literature [32]. For this purpose, we have followed the same scheme as in IS, which has also been previously applied with success [34]. We consider the CHC algorithm to find the best feature (classifier) subset, performing a CS via FS. Once again we consider a binary chromosome to encode the solution (right side of the scheme in Fig. 1), where each gene (c_{x_i}) indicates whether a classifier/feature is selected (1) or not (0):

$$C_{CS} = (c_{x_1}, c_{x_2}, c_{x_3}, c_{x_4}, \ldots, c_{x_L}), \tag{11}$$

where L is the number of classifiers, that is, $m(m-1)/2$.

In order to evaluate each chromosome, we can also focus only on accuracy, but it is interesting to force the reduction of classifiers if possible. Hence, we also consider another fitness function only based on accuracy.

3. **AccRedCS (f_{AS})**: A lexicographical order is established. Accuracy is taken into account first. If there is a tie, the chromosome with the greatest classifier reduction is selected.

The computation of the fitness function is done in the same manner as in IS case.

4.4. Combining instance and feature selection

Finally, it can be easily observed that both models can be combined and performed simultaneously looking for the feature and instance subset with the best interaction. With this aim, both chromosomes are considered as only one where genes are concatenated and the evolution process is performed in the same manner.

In this case, we can also mix the previous fitness functions to analyzed whether forcing instance or classifier reductions hinder accuracy or even if they can help improving the generalization ability. Therefore, two more fitness functions are considered:

3. **AccRedCSRedIS (f_{ACI})**: A lexicographical order is established. Accuracy is taken into account first. If there is a tie, the chromosome with the greatest classifier reduction is selected. If the tie still continues, the best chromosome is decided with respect to instance reduction.
4. **AccRedISRedCS (f_{AIC})**: A lexicographical order is established. Accuracy is taken into account first. If there is a tie, the chromosome with the greatest instance reduction is selected. If the tie still continues, the best chromosome is decided with respect to classifier reduction.

4.5. CHC genetic algorithm

As we have already mentioned, we make use of the well-known CHC algorithm due to its good behavior in the topic [34,35,55,56], which leaves the study of the behavior of other models as well as the usage of different fitness functions for future works. In CHC algorithm, all the M chromosomes in the population and their offspring (obtained by the crossover operator) are put together; then, the next population is formed of the M best individuals (in terms of the fitness function considered). In this GA, instead of using a mutation operator as other GAs do, an incest prevention mechanism combined with a re-initialization of the population is used to promote diversity (as explained hereafter). The rest of the necessary components to design the whole process are: initialization of the initial population, crossover operator, incest prevention and restarting mechanisms, given that the representation of the solutions (encoding of the chromosomes) have been previously explained for each case.

1. *Initial population:* The initial population is formed of random chromosomes except for one that is taken to have all its genes set to 1 in order to represent the original model (without any selection).
2. *Crossover Operator:* The Heterogenous Uniform Crossover (HUX) is used, since we are considering binary chromosomes. This operator interchanges exactly half of the different genes between both individuals selected. In the case of its application to IS and CS, the original HUX is modified, decreasing the probability of including instances or features. In this manner, each time HUX switches a gene on, the gene is switched off with a certain probability (whose recommended value is 0.25).
3. *Incest prevention:* The crossover between too similar parent are prevented, that is, between parents having their Hamming distance (divided by two) below a threshold value T. This threshold is initially established to be $N_c/4$ being N_c the length of the chromosome. If no individuals are recombined, then the threshold value is reduced by one.
4. *Restarting mechanism:* The mutation operator is replaced by this mechanism in CHC aiming at avoiding local optima. When the threshold value T reaches a zero value, all the chromosomes in the population, except for the best one (following an elitist scheme), are eliminated and generated again. New chromosomes are created by randomly changing 35% of the genes of the best chromosome. As in the crossover operator, the likelihood of including instances or features is decreased by giving less probability to set a gene on than to set it off (the same value as before is used, 0.25).

The optimization process is finished when any of the following stopping criteria are met: the number of evaluations or the number of restarting procedures without improvements reach their maximum values. Their set-up is detailed in the experimental framework (Section 5.1). Finally, the whole training set is used to carry out the optimization procedure.

5. Experimental framework

In this section, the set-up of the experimental framework used to develop the empirical comparison in Section 6 is introduced. The base classifiers considered and their configuration are described first (Section 5.1). Afterwards, the best combinations for each base classifier [18] that will be the baseline for the comparisons as explained in Section 2.2 together with the configuration for the DCS approach [24] are recalled (Section 5.2). Next, details of the data-sets are given (Section 5.3), and the statistical tests used to make the comparison are explained (Section 5.4). Finally, details on the methodology followed throughout the experimental study are given (Section 5.5).

5.1. Base learners and parameter configuration

In order to show the usefulness of our combination for the OVO strategy allowing for classifier pruning, we have selected several well-known Machine Learning algorithms as base learners:

- ***k*NN** – *k-Nearest Neighbors* [29].
- **SVM** – *Support Vector Machine* [5].
- **C4.5** – decision tree [37].
- **Ripper** – *Repeated Incremental Pruning to Produce Error Reduction* [39].

These learning algorithms were selected due to their good behavior in a large number of real-world problems. Moreover, in case of SVM there is not a multi-category approach established yet, although there are several attempts [57].

The majority of the combination methods in OVO make use of the confidence degrees of the outputs of each base classifier. These confidence degrees are obtained for each classifier as follows:

Table 1
Parameter specification for the base learners employed in the experimentation and CHC algorithm.

Algorithm	Parameters
3NN	$k = 3$, Distance metric = HVDM
SVM_{Poly}	C = 1.0, Tolerance Parameter = 0.001, Epsilon = 1.0E-12
	Kernel Type = Polynomial, Polynomial Degree = 1
	Fit Logistic Models = True
SVM_{Puk}	C = 100.0, Tolerance Parameter = 0.001, Epsilon = 1.0E-12
	Kernel Type = Puk, PukKernel ω = 1.0, PukKernel σ = 1.0
	Fit Logistic Models = True
SVM_{Fit}	C = {0.01, 0.1, 1.0, 10.0, 100.0, 1000.0}, Tolerance Parameter = 0.001
	Epsilon = 1.0E-12, Kernel Type = {Polynomial, RBF, Puk}
	Polynomial Degree = 1, RBF γ = {0.001, 0.01, 0.1, 1}
	PukKernel ω = 1.0, PukKernel σ = 1.0, Validation = 5-fold cross-validation
	Fit Logistic Models = True
C4.5	Prune = True, Confidence level = 0.25
	Minimum number of item-sets per leaf = 2
Ripper	Size of growing subset = 66%, Repetitions of the optimization stage = 2
CHC	Population size (M) = 50 individuals, Evaluations = 40000
	Restarting procedures without improvement = 3

- *k* **NN –** Distance-based confidence estimation.

$$Confidence = \frac{\sum_{l=1}^{k} \frac{e_l}{d_l}}{\sum_{l=1}^{k} \frac{1}{d_l}} \tag{12}$$

  where d_l is the distance between the input pattern and the lth neighbor and $e_l = 1$ if the neighbor l is from the predicted class and 0 otherwise. Note that when $k > 1$, the probability estimate depends on the distance from the neighbors, hence the estimation is not restricted to a few values.
- **SVM –** Probability estimates obtained by the SVM logistic model [58].
- **C4.5 –** Accuracy of the leaf making the prediction, i.e., the number of correctly classified training examples divided by the total number of covered train instances.
- **Ripper –** Accuracy of the rule used in the prediction (the same computation as in C4.5 considering rules instead of leafs).

In some of the combination strategies ties might occur. As usual, in those cases the majority class is predicted. If the tie continues, the class is selected randomly.

The parameters used in each base classifier are shown in Table 1, which also includes the parameters of the CHC algorithm. These values are common for all problems, and they were selected according to the recommendation of the corresponding authors, which is also the default setting of the parameters included in the KEEL[3] software [11,36], which we have used to develop our experiments. In the case of SVMs, three configurations are considered. In the first two ones, the parameter C and the kernel function are fixed in order to study the behavior of our strategy with different configurations, which should address for the robustness of the proposal (in the sense that despite how fine-tuned are the base classifiers, its behavior is maintained with respect to the others). In the third configuration, we considered a fine-tuned SVM (with respect to the accuracy obtained by OVO using an internal 5-fold cross-validation scheme). In this way, we are able to analyze the behavior of NMC with highly fitted classifiers with different parameters for each data-set and with a little room for improvement. We treat nominal attributes in SVM as scalars to fit the data into the systems using a polynomial kernel.

We acknowledge that the tuning of the parameters in all the classifiers for each particular problem could lead to better results, however, we preferred to maintain a baseline performance on each method as the basis for comparison. Since we are not comparing base classifiers among each other, our hypothesis is that the methods (combinations) that win on average on all problems would also win if a better setting is performed. Moreover, when methods are not tuned, winner methods tend to correspond to the most robust ones, which is also desirable. Anyway, given that in SVMs tuning can lead to highly improved results and it is a commonly considered process, we have also considered SVM_{Fit} in order to show that the usefulness of the method remain unchanged in this scenario.

5.2. Combinations considered

We use a different combination for each base classifier to analyze their behavior in comparison with our proposal since the best combination model depends on the base classifier. For this reason, we follow our findings from our previous work [18] and use the same representatives for each base classifier as those selected in it (the best ones).

The following combinations are considered:

- ***k*NN** – ND (Non-Dominance criterion).
- **SVM** – PE (Wu et al. [23] Probability Estimates by Pairwise Coupling).
- **C4.5** – WV (Weighted Voting strategy).
- **Ripper** – WV (Weighted Voting strategy).

In addition, we have also considered the DCS approach [24], which outperformed most of them in the same experimental framework as the one we are considering. We use the same parameter value for k as the one in the original DCS paper, i.e., $k = 3 \cdot m$ (3 times the number of classes) is considered as the neighborhood to select competent classifiers (notice that this k is not the one used in our proposal).

5.3. Data-sets

We have used twenty-three data-sets from the KEEL data-set repository[4] [36]. Data-sets with a large representation of different number of classes and attributes have been considered. Their properties are summarized in Tables 2 and 3. In the former table for each data-set, the number of examples (#Ex.), the number of attributes (#Atts.), the number of numerical (#Num.) and nominal (#Nom.) attributes, and the number of classes (#Cl.) are shown. In the latter one, the number of instances from each class in each

[3] http://www.keel.es.

[4] http://www.keel.es/dataset.php.

Table 2
Summary description of data-sets.

Data-set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.
Balance	625	4	4	0	3
Hayes-roth	132	4	4	0	3
Iris	150	4	4	0	3
NewThyroid	215	5	5	0	3
Splice	319	60	0	60	3
Tae	151	5	5	0	3
Thyroid	720	21	21	0	3
Wine	178	13	13	0	3
Car	1728	6	0	6	4
Lymphography	148	18	3	15	4
Cleveland	297	13	13	0	5
Nursery	1296	8	0	8	5
Page-blocks	548	10	10	0	5
Shuttle	2175	9	9	0	5
Autos	159	25	15	10	6
Dermatology	358	34	1	33	6
Glass	214	9	9	0	7
Satimage	643	36	36	0	7
Segment	2310	19	19	0	7
Zoo	101	16	0	16	7
Ecoli	336	7	7	0	8
Penbased	1100	16	16	0	10
Vowel	990	13	13	0	11

data-set is presented. As it can be observed, they comprise a number of situations, from totally balanced data-sets to highly imbalanced ones, besides the different number of classes.

The performance estimates were obtained by means of a 5-fold stratified cross-validation (SCV). From our point view, 5-fold SCV is more appropriate than a 10-fold SCV in the current framework, since using smaller partitions there would be more test sets that will not contain any instance from some of the classes. More specifically, the data partitions were obtained by the Distribution Optimally Balanced SCV (DOB-SCV) [59], which aims to correct the data-set shift (when the training data and the test data do not follow the same distribution) that might be produced when dividing the data. In order to address the stochastic nature of GAs, the 5-fold SCV is repeated five times with different seeds.

5.4. Statistical tests

In order to assess the results obtained by each model, we have considered the accuracy rate as performance measure. Additionally, we will consider the reduction rate of instances and features (number of score matrices and binary classifiers, respectively) together with the training times spent in obtaining such reduced sets. Testing times have also been computed, but we have observed that the maximum testing time per instance is much lower than one millisecond (almost 10 times lower in most data-sets). Such a fast testing times show the practical utility of the model, and the fact that it does not cause a significant increase in the testing times. For this reason and for the sake of brevity, we have not included the detailed testing times in the paper and we focus on the cost of the learning stage. Finally, experiments have been carried out under a computer with an Intel(R) Core(TM) i7 CPU 930 microprocessor (4 cores/8 threads, 2.8 GHz, 8 MB Cache) with 24 GB of DDR2 RAM memory and using CentOS 6.4. The maximum Java heap space reserved for each execution was only 1GB.

In order to make a fair comparison of the performance of the classifiers, we perform the corresponding statistical analysis as recommended in the literature [40,41]. Hence, non-parametric statistical tests are considered (for more information please refer to http://sci2s.ugr.es/sicidm/).

Different types of comparisons are carried out in the experimental study. When multiple methods are compared, we use Friedman aligned-ranks test [60] as a non-parametric statistical procedure to perform comparison among a set of algorithms. Then, if this test detects significant differences among them, we check if the control algorithm (the best one) is significantly better than the others (that is, $1 \times n$ comparison) using Holm post-hoc test [61].

Moreover, we consider the average aligned-ranks of each algorithm (used in the Friedman aligned-ranks test) in order to compare the behavior of each algorithm with respect to the others. These rankings are obtained computing the difference between the performance obtained by the algorithm and the mean performance of all the algorithms in the corresponding data-set. These differences are ranked from 1 to $k \cdot n$ (being k the number of data-sets and n the number of methods), assigning the corresponding rank to the method from which the difference has been computed. Hence, the lower the rank is, the better the method is. At last, the

Table 3
Number of instances per class in each data-set.

Data-set	#Ex.	#Cl.	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
Balance	625	3	288	49	288								
Hayes-roth	132	3	51	51	30								
Iris	150	3	50	50	50								
NewThyroid	215	3	30	35	150								
Splice	319	3	77	77	165								
Tae	151	3	49	50	52								
Thyroid	720	3	17	37	666								
Wine	178	3	59	71	48								
Car	1728	4	1210	384	65	69							
Lymphography	148	4	2	81	61	4							
Cleveland	297	5	160	54	35	35	13						
Nursery	1296	5	1	32	405	426	432						
Page-blocks	548	5	492	33	8	12	3						
Shuttle	2175	5	1706	2	6	338	123						
Autos	159	6	3	20	48	46	29	13					
Dermatology	358	6	111	60	71	48	48	20					
Glass	214	7	70	76	17	0	13	9	29				
Satimage	643	7	154	70	136	62	71	0	150				
Segment	2310	7	330	330	330	330	330	330	330				
Zoo	101	7	41	20	5	13	4	8	10				
Ecoli	336	8	143	77	2	2	35	20	5	52			
Penbased	1100	10	115	114	114	106	114	106	105	115	105	106	
Vowel	990	11	90	90	90	90	90	90	90	90	90	90	90

average ranking of each algorithm in all data-sets can be computed to show their global performance.

Additionally, we consider the Wilcoxon paired signed-rank test [62] as a non-parametric statistical procedure when we need to perform comparisons between two methods.

5.5. Methodology of analysis

Given the large amount of combination tested in the experimental study, they are first outlined in this section. We should recall that for each base classifier considered, we include the following methods in the comparison.

1. **Comb**: The best combination/aggregation method for the base classifier (see Section 5.2).
2. **DCS**: The combination based on DCS for the OVO strategy [24].
3. **WV**: An adaptation of the original WV method, which uses the arithmetic mean of the outputs for the class instead of the sum. Hence, the result is equivalent to the original WV when no classifiers are removed, but it allows us to consider the removal of classifiers using this strategy (since it is no longer affected by the numbers of elements that are summed up). As a result, we have the original WV (**NoGA**) and *WV with classifier pruning* (**CS**), which uses the same GA as NMC.
4. **NMC**: Our proposal based on the usage of kNN for combining classifiers. In this study, $k = 1$ is considered, even though we should emphasize the fact that preliminary results were equivalent with different k values. We have four possible models in this case.
   - **NoGA**: NMC is applied using all the instances and classifiers.
   - **CS**: Only classifier (feature) selection is considered.
   - **IS**: Only instance selection is considered.
   - **CS_IS**: Both instance and classifier selections are performed simultaneously.
5. **DT**: Decision templates where one template per class is computed (see Section 2.3). In this case, we have also added the possibility of reducing the number of classifiers with the same GA as NMC (**CS**), maintaining also the original one named as **NoGA**.

In those cases where classifier selection is performed (WV, NMC and DT) two fitness functions are considered ($\boldsymbol{f_A}$ and $\boldsymbol{f_{AC}}$). In the case of NMC, we also consider the fitness functions $\boldsymbol{f_{AI}}$, $\boldsymbol{f_{AIC}}$ and $\boldsymbol{f_{ACI}}$ in the corresponding cases. Recall that all the fitness functions that forcing instance or classifier selection considers a lexicographical order where accuracy is taken into account first and better reductions are ranked higher in the case of ties.

In the cases of WV and DTs with CS, the CHC algorithm is used exactly as in the case of NMC except for the computation of the outputs, which is made using WV and DTs, respectively, taking only the classifiers selected into account.

6. Experimental study

In this section our aim is to evaluate the usefulness of our proposal. For each model, we obtained the results of accuracy, classifier reduction rate (**RedCS**), instance reduction rate (**RedIS**) and training times in the cases where the GA is executed. We should recall that testing times are not included since all of them are below 1 millisecond for each instance (in all of the methods and data-sets).

On account of all these possibilities we have divided the statistical analysis carried out into four phases:

- **Phase 1:** *Study of the best fitness function.* We study the effect of the different fitness functions that do not only take accuracy into account. We will show the fact that CS in WV does not work properly, different from our model and contrary to the case of DTs, where it is mandatory in order to be competitive (Section 6.1).
- **Phase 2:** *Study of the best NMC model.* We analyze the results of the different models from our proposal looking at their advantages and disadvantages (Section 6.2).
- **Phase 3:** *NMC vs. WV (without pruning) vs. DTs.* We test the behavior of the three models allowing for CS, which in the case of WV does not work (Section 6.3).
- **Phase 4 – Final:** *NMC vs. state-of-the-art combinations (Comb and DCS).* We show that NMC can be competitive with the state-of-the-art combinations, with the additional advantage of being able to reduce the number of classifiers (Section 6.4).
- **Discussion:** At last, the main points of this study are summarized in Section 6.5.

Before starting with the analysis, we show the overall accuracy rates in test, instance and classifier reduction rates and training times in Table 4. The detailed results of all the methods and data-sets are presented in Appendix A.

Along this section we use the two statistical tests mentioned. They will be presented as follows:

- *Friedman aligned-rank tests*: It is used when more than two methods are compared. A test is carried out for each base classifier. An example can be observed in Table 5. In these tables, the aligned-ranks obtained by each method (row) in each base classifier (column) are shown (the lower the better), that is, a test is run for each column. Near the ranks obtained by each method the p-value given by the Holm post-hoc test is shown in brackets, which compares a control method (the best one, i.e., the one with the lowest rank in a column) against the rest.
- *Wilcoxon tests*: It is used to compare a pair of methods. A test is also performed for each base classifier (column), as it can be observed in Table 6. For each comparison the ranks obtained by each method are presented in the first row (the greater the better) and in the second row the p-value associated with the comparison is shown.

In both tables a '+' close to the p-value means that statistical differences are found in the comparison with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence).

6.1. Study of the best fitness function

In this Section we analyze the behavior of the different fitness functions considered, for each one of the models in which they are used: WV, NMC (CS, IS, CS_IS) and DTs. This analysis is carried out separately for each model.

- *WV*. Two fitness functions are considered for the WV with CS (f_A and f_{AC}). Looking at Table 4, one can observe that accuracy is almost the same in both cases except for the case of SVM_{Puk}. However, even though the absolute differences are rather small there is a tendency to decrease accuracy when CS is used, and more if it is forced. Anyway, these almost equal global accuracies can be explained by the fact that classifier reduction is almost non-existent with f_A, being 3NN and SVM_{Poly} the cases with the greatest reduction with 10% and 5%, respectively. When forcing reduction (f_{AC}), these rates are increased around 5%, with the greatest improvement in SVM_{Puk} (where indeed the largest loss of accuracy is produced). The real differences among these models can be better analyzed following the statistical tests shown in Table 5. Statistical differences are only found in the case of SVM_{Puk} using f_{AC}. Hence, when reduction is forced to some extent accuracy is affected. In the rest of the cases, whenever some reduction has been achieved accuracy also drops, but due to the low reduction rates obtained

Table 4
Average accuracy results in test, classifier and instance reduction rates and training times for the six classifier tested. The best result in each row is highlighted in bold-face.

				WV			NMC with $k=1$										DT		
					CS			CS		IS		CS_IS						CS	
Classifier	Measure	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
3NN	Accuracy	86.34	86.93	86.21	85.93	85.93	82.55	82.15	81.92	86.91	86.81	86.86	86.86	**86.96**	86.90	86.74	83.44	85.85	85.81
	RedCS	–	–	–	9.59	13.78	–	31.90	43.43	–	–	41.89	**50.14**	46.63	48.58	49.95	–	37.86	47.39
	RedIS	–	–	–	–	–	–	–	–	75.29	91.82	77.74	80.86	92.58	92.67	92.45	–	**97.97**	**97.97**
	Time	–	–	–	1.00	**0.98**	–	170.8	186.0	172.5	113.1	151.7	143.2	109.3	110.2	113.2	–	1.71	1.66
SVM_{Poly}	Accuracy	84.16	84.24	84.09	83.95	83.98	**86.43**	85.90	85.82	85.89	85.73	86.07	85.95	85.81	85.84	85.81	81.32	83.04	83.00
	RedCS	–	–	–	4.89	9.92	–	36.13	50.20	–	–	44.31	**54.19**	48.96	51.45	53.97	–	34.08	50.02
	RedIS	–	–	–	–	–	–	–	–	68.13	91.18	69.52	75.90	91.57	91.57	91.12	–	**97.97**	**97.97**
	Time	–	–	–	0.85	**0.80**	–	196.2	207.1	254.9	215.8	231.8	216.9	200.7	195.1	206.9	–	1.66	1.66
SVM_{Puk}	Accuracy	83.23	**85.21**	83.49	83.49	82.44	84.79	84.56	82.57	84.35	84.30	84.53	82.20	83.79	83.73	83.22	82.62	83.91	83.62
	RedCS	–	–	–	0.00	16.49	–	18.01	53.05	–	–	20.58	52.85	42.79	47.85	**53.41**	–	22.60	45.17
	RedIS	–	–	–	–	–	–	–	–	34.80	93.75	38.41	67.25	94.82	94.84	94.56	–	**97.97**	**97.97**
	Time	–	–	–	0.87	**0.84**	–	125.3	147.5	337.3	103.7	256.1	181.5	83.81	84.39	88.93	–	1.82	1.78
SVM_{Fit}	Accuracy	89.53	89.59	89.72	89.67	**89.76**	89.45	89.18	88.95	89.35	89.27	89.30	89.08	89.10	89.09	89.04	87.14	89.15	89.04
	RedCS	–	–	–	1.34	7.94	–	30.55	50.68	–	–	32.59	**50.70**	44.66	47.11	50.52	–	27.27	43.48
	RedIS	–	–	–	–	–	–	–	–	55.43	93.56	59.10	73.61	94.33	94.46	94.14	–	**97.97**	**97.97**
	Time	–	–	–	1.78	1.68	–	152.8	162.0	414.9	158.3	246.9	181.0	100.8	100.2	104.3	–	1.38	**1.24**
C4.5	Accuracy	85.35	85.82	85.35	85.33	85.33	81.90	81.80	81.65	**85.87**	85.83	85.51	85.41	85.55	85.50	85.45	82.65	84.11	84.06
	RedCS	–	–	–	0.14	5.63	–	22.09	43.83	–	–	33.94	**46.46**	42.80	44.86	45.62	–	25.77	46.05
	RedIS	–	–	–	–	–	–	–	–	62.55	93.10	64.65	71.35	93.62	93.76	93.32	–	**97.97**	**97.97**
	Time	–	–	–	0.98	**0.97**	–	198.6	220.3	231.7	126.8	200.9	170.7	115.4	113.9	121.3	–	2.33	2.66
Ripper	Accuracy	85.29	85.58	85.26	85.25	85.21	**85.89**	85.22	85.19	85.80	85.70	85.59	85.49	85.56	85.54	85.51	84.05	84.85	84.85
	RedCS	–	–	–	0.38	2.05	–	22.98	38.00	–	–	28.56	40.64	36.85	39.18	**40.92**	–	25.11	38.96
	RedIS	–	–	–	–	–	–	–	–	59.50	92.70	62.05	72.41	93.15	93.10	93.06	–	**97.97**	**97.97**
	Time	–	–	–	1.08	**1.06**	–	202.0	203.5	206.4	122.2	193.2	168.7	120.9	119.5	123.7	–	2.29	2.33

Table 5
Friedman aligned-rank tests comparing the different WV models in each base classifier with accuracy.

Methods	3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
WV	32.24	29.30	31.11	34.85 (0.42097)	34.74	31.17
WV with CS-f_A	35.70 (0.82928)	39.07 (0.19793)	31.11 (1.00000)	40.07 (0.18335)	35.00 (1.00000)	34.02 (0.63025)
WV with CS-f_{AC}	37.07 (0.82928)	36.63 (0.21559)	42.78 (0.09693+)	30.09	35.26 (1.00000)	39.80 (0.28923)

A '+' means that there are statistical differences with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence)

Table 6
Wilcoxon tests to compare f_A with f_{AC} in NMC with CS using accuracy.

Comparison		3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
f_A vs. f_{AC}	R^+/R^-	201.5/74.5	155.5/120.5	220.5/55.5	172.0/104.0	198.5/77.5	139.0/137.0
	p-value	0.02194*	0.49246	0.01571*	0.29588	0.03546*	0.77643

R^+ corresponds to the sum of the ranks for $\text{NMC}^{CS}_{f_A}$ and R^- for $\text{NMC}^{CS}_{f_{AC}}$.

significant differences are not found. The only exception is the case of SVM_{Fit} where forcing seems to perform well, but looking at the accuracy from Tables 4 and A.18, it can be observed that the real differences between all the approaches are below 0.1% due to the highly adjusted base classifiers. Hence, few classifiers can be safely removed using the WV, being this reduction meaningless in most of the cases. Moreover, accuracy tends to decrease as classifiers are removed. This result is in accordance with [51], where OVO lost precision linearly as classifiers where removed. However, it can be observed that selecting the proper aggregation, the linear decrease can be avoided when removing some of the classifiers, even though after the 10% (as in SVM_{Puk}) significant differences appear. On this account we consider the original WV for the next comparisons given that it is the best performer.

- *NMC with CS*. In this case f_A and f_{AC} fitness functions are considered. Looking at Table 4, it can be observed that forcing classifier reduction can lead to reduction rates between 40% and 50%, whereas without doing so, reductions between 20% and 30% are obtained. The question is whether this greater reduction comes along with a drop of accuracy, whose answer varies depending on the base classifier. As the Wilcoxon test in Table 6 indicates, in 3NN, SVM_{Puk} and C4.5 statistical difference are found in favor of not forcing reduction, whereas in SVM_{Poly}, SVM_{Fit} and Ripper these differences are not significant. Anyway, since our ultimate objective is to maintain accuracy while reducing classifiers, we consider f_A function, although one could prefer a greater classifier reduction in which case f_{AC} function would be recommended.
- *NMC with IS*. f_A and f_{AI} are compared in this case. In terms of reduction, it is clear from Table 4 that promoting IS helps in reducing the final size of the reference set for NMC. Between 15% and 30% more instances (score-matrices) can be removed, even 60% more in the case of SVM_{Puk}, which shows great differences with respect to the behavior of SVM_{Poly} as in previous works [25,56] due to the probabilities obtained being too crisp as a consequence of the configuration considered (near 0 or 1). Tuned SVMs (SVM_{Fit}) behave similarly to SVM_{Poly}. Even though in global terms there seems not to be differences in terms of accuracy between these functions, we performed the corresponding Wilcoxon tests, which are shown in Table 7. From them, it can be concluded that accuracy is not statistically hindered when IS is forced, but the results get worse in all the classifiers, being the ranks always in favor of f_A. Again, in order to maintain accuracy as high as possible, we consider f_A, although f_{AI} can be useful to reduce the reference set in large data-sets.

Table 7
Wilcoxon tests to compare f_A with f_{AI} in NMC with IS using accuracy.

Comparison		3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
f_A vs. f_{AI}	R^+/R^-	159.0/117.0	174.0/102.0	167.5/108.5	178.5/97.5	175.0/101.0	170.5/105.5
	p-value	0.52301	0.27354	0.39446	0.21724	0.40804	0.33916

R^+ corresponds to the sum of the ranks for $NMC^{IS}_{f_A}$ and R^- for $NMC^{IS}_{f_{AI}}$.

Table 8
Friedman aligned-rank tests comparing the different fitness functions for NMC with CS and IS in each base classifier with accuracy.

Methods	3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
f_A	58.93 (1.00000)	39.54	50.39	43.41	53.59	55.43 (1.00000)
f_{AC}	58.61 (1.00000)	54.26 (0.13442)	60.04 (0.77281)	64.67 (0.12234)	67.70 (0.60516)	63.54 (1.00000)
f_{AI}	52.13	65.72 (0.03106*)	63.22 (0.76820)	60.70 (0.17170)	56.89 (1.00000)	53.11
f_{AIC}	54.46 (1.00000)	65.57 (0.03106*)	54.83 (0.77281)	59.11 (0.17170)	55.24 (1.00000)	57.00 (1.00000)
f_{ACI}	65.87 (0.64917)	64.91 (0.03106*)	61.52 (0.77281)	62.11 (0.17170)	56.59 (1.00000)	60.91 (1.00000)

A '+' means that there are statistical differences with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence)

Table 9
Wilcoxon tests to compare f_A with f_{AI} in NMC with CS and IS using accuracy.

Comparison		3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
f_A vs. f_{AI}	R^+/R^-	114.0/162.0	229.5/46.5	212.5/63.5	196.5/79.5	143.0/133.0	140.5/135.5
	p-value	0.46542	0.00551*	0.02209*	0.07683+	0.77818	0.93531

R^+ corresponds to the sum of the ranks for $NMC^{CIS}_{f_A}$ and R^- for $NMC^{CIS}_{f_{AI}}$.

- *NMC with CS and IS.* When performing both CS and IS simultaneously we have to analyze the whole set of fitness functions, including those forcing IS and CS at the same time. Looking at Table 4, it can be observed that the mere fact of carrying out both processes simultaneously allows one to achieve better results in terms of accuracy and reduction rates in all the base classifiers considering the same fitness function f_A than when they are performed individually. Hence, IS allows for a better CS and vice versa. At first glance, the results of the fitness functions forcing IS (f_{AI}, f_{AIC}, f_{ACI}) are similar for all the measures. Only forcing CS (f_{AC}) usually leads to more classifiers being removed (only reducing accuracy in the case of SVM_{Puk}), and even more instances than when only focusing on accuracy (f_A). In order to gain a better insight of these results, we performed the statistical tests shown in Tables 8 and 9. The first one presents the Friedman aligned-rank tests for each base classifier and all the fitness functions, whereas the second one shows the direct comparison between f_A and f_{AI}, which are the best performers in the first analysis.
  Looking at these tables we can observe that in terms of accuracy the best options are the ones not forcing CS in any way, but forcing IS does not have the same effect, achieving the lowest ranks in 3NN and Ripper. On this account, we carry out the Wilcoxon test to compare these alternatives, from which it can be concluded that only focusing on accuracy may be preferable if one wants to reach the highest accuracy as possible in a general manner (with any classifier). In fact, in the case of 3NN it is beneficial to force IS, but in the case of SVMs, statistical differences are found against this alternative. Therefore, we continue with f_A, even though f_{AI} could become an interesting model in data-sets with a large number of instances.
- *DT*. Like in WV, we allowed DTs to perform CS, and hence two different fitness functions can be considered (f_A and f_{AC}). However, in this case the results obtained are different to those of WV. Accuracy is boosted in all cases when CS is considered, and even forcing CS in the fitness function seems not to affect accuracy, while almost 50% of the classifiers are reduced. In order to contrast this fact, we have performed the Friedman aligned-rank tests (Table 10), where the superiority of the CS models is clearly observed. Consequently, we have confronted both possible fitness functions to decide which is better suited for DTs using the Wilcoxon test (Table 11). According to the results of these tests, there are no statistical differences with any of the classifiers. In two cases f_{AC} obtains more ranks than f_A (3NN and SVM_{Puk}), and in three times f_A beats f_{AC} (SVM_{Poly}, SVM_{Fit} and Ripper), whereas in one case there is a tie (C4.5). On this account, we select the f_{AC} fitness function since it obtains better reductions (between 40% and 50%) without losing accuracy and given the good synergy between DTs and pruning.

Remark 2. Summing up, we have shown that WV does not allow for a meaningful CS, whereas DTs basically require it. Regarding NMC, the different fitness functions allow one to play with the trade-off between reduction and accuracy. Focusing in the latter, we have considered f_A for the next sections, even though forcing IS sometimes can help increasing accuracy. Overall, it can be observed that around 40–50% of the classifiers can be safely removed from OVO without loosing accuracy with respect to NMC with IS. In fact, it should be mentioned that NMC by itself, or NMC with CS, do not perform competitively in some cases, and hence considering IS is required as we show in the next section.

6.2. Study of the best NMC model

We are looking for the best NMC model in this section. Looking at Table 4, the varying behavior of NMC without selection (NoGA) in each base classifier is clear. In 3NN and C4.5, these results and the ones of CS are clearly inferior to those in which IS is performed, showing that IS is mandatory to remove the noise produced by some of the instances. Otherwise, in the rest of the classifiers, NMC by itself achieves an excellent performance with the best results in SVM_{Poly} and Ripper. However, since we are more interested in removing classifiers and in a model which is able to work with any base classifier, we should focus on CS_IS model. The accuracy of this model is competitive with IS attending at the table of results, including 3NN and C4.5, overcoming the problems of NMC without further processing. Anyway, we should base all these conclusions in the proper statistical analysis, which is carried out in Table 12.

The results in Table 12 follow our previous claims. NMC without selection (NoGA) is the best model for some of the base classi-

Table 10
Friedman aligned-rank tests comparing the different DTs models in each base classifier with accuracy.

Methods	3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
DT	53.02 (0.00001*)	46.28 (0.00487*)	39.89 (0.32744)	47.15 (0.00027*)	43.72 (0.05047+)	44.78 (0.01578*)
DT with CS-f_A	25.59	28.35	33.46 (0.76037)	24.54	30.48	29.07
DT with CS-f_{AC}	26.39 (0.89185)	30.37 (0.73255)	31.65	33.30 (0.13864)	30.80 (0.95604)	31.15 (0.72427)

A '+' means that there are statistical differences with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence)

Table 11
Wilcoxon tests to compare f_A and f_{AC} in DTs with CS using accuracy.

Comparison		3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
f_A vs. f_{AC}	R^+/R^-	129.0/147.0	173.0/103.0	114.5/161.5	172.5/103.5	138.0/138.0	152.5/123.5
	p-value	0.87533	0.39425	0.49246	0.28598	1.00000	0.79459

R^+ corresponds to the sum of the ranks for $DT^{CS}_{f_A}$ and R^- for $DT^{CS}_{f_{AC}}$.

Table 12
Friedman aligned-rank tests comparing the different NMC models selected (CS, IS and CS_IS with f_A) in each base classifier with accuracy.

Methods	3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
NoGA	55.54 (0.00818*)	41.24	41.85	44.67 (1.00000)	56.48 (0.00078*)	37.02
CS-f_A	62.46 (0.00053*)	50.87 (0.66391)	53.74 (0.39298)	51.80 (0.71863)	61.30 (0.00010*)	58.26 (0.02097*)
IS-f_A	32.93	48.54 (0.70718)	42.46 (0.93838)	42.54	28.54	41.89 (0.53629)
CS_IS-f_A	35.07 (0.78673)	45.35 (0.70718)	47.96 (0.87573)	46.98 (1.00000)	39.67 (0.15749)	48.83 (0.26767)

A '+' means that there are statistical differences with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence)

Table 13
Friedman aligned-rank tests comparing WV, NMC with IS and CS using f_A and DTs with CS using f_{AC} in each base classifier with accuracy.

Methods	3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
WV	36.80 (0.07472+)	35.87 (0.00662*)	38.37 (0.17776)	25.72	30.09 (0.53460)	33.83 (0.29329)
NMC with CS_IS-f_A	26.26	19.80	28.30	36.85 (0.05992+)	26.41	27.61
DT with CS-f_{AC}	41.93 (0.01613*)	49.33 (0.00000*)	38.33 (0.17776)	42.43 (0.00943*)	48.50 (0.00038*)	43.57 (0.01399*)

A '+' means that there are statistical differences with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence)

fiers, but it is statistically outperformed by the IS procedure in two cases (3NN and C4.5), whereas the contrary does not occur. Nevertheless, none of them is capable of reducing classifiers. Whereas CS model is statistically outperformed in three out of five base classifiers, no significant differences are found against CS_IS model, which uses the least amount of classifiers. Therefore, the good capabilities shown by the CS_IS approach allow us to consider it for further comparisons.

6.3. NMC vs. WV (without pruning) vs. DTs

This section is devoted to compare the different methods in which we have considered CS, that is, WV, NMC and DTs. Recall that in the case of WV, CS has a negative effect, and hence the original WV method is compared. In the case of NMC, we continue with CS_IS using f_A and in DTs with CS using f_{AC}. Therefore, it has to be taken into account that NMC and DTs are being compared versus WV, one of the most robust alternatives for aggregating classifiers in OVO [14]. Clearly, WV does not reduce classifiers, so it should have an advantage. Between NMC and DTs, the latter is the one achieving the largest reduction rate in terms of classifiers and instances (it only uses one instance per class). However, this greater reduction makes it achieve lower average accuracy rates (notice that DTs without pruning obtains an even lower accuracy). The Friedman aligned-rank tests to compare these methods in all the classifiers are presented in Table 13.

Looking at the results of these tests, it can be observed that removing classifiers can lead to better results with respect to previous aggregation strategies such as the WV in most of the classifiers. NMC is the one achieving the lowest ranks, i.e., the best performance, in all the base classifiers except for SVM_{Fit}. Otherwise, DTs, also allowing for CS, are significantly outperformed by NMC in four out of six classifiers. With respect to WV, statistical differences are found with 3NN and SVM_{Poly}; the p-value for SVM_{Puk} is also low, whereas the differences in C4.5 and Ripper are not high in terms of p-values. The case of SVM_{Fit} can be easily explained looking at the differences in the complete table of results presented in the Appendix A (Table A.18). Average accuracy difference between WV and NMC is as low as 0.42%, but it resulted in a rejection due to the fact that WV is constantly better than NMC. Nonetheless, the absolute difference between both models could make NMC useful given the fact that it allows for classifier pruning (around 30% of the classifiers are removed). This behavior appears due to the highly fine-tuned base classifiers, which make almost all the results of SVM_{Fit} to be in less than 1% of accuracy difference.

We conclude that the different behavior among the base classifiers can be justified by their ability to give good confidence degrees and their base performance. In this respect, SVM_{Poly} using the logistic model is the one giving the best confidence degrees, whereas the same does not occur with SVM_{Puk} due to the different configuration. As we have explained, SVM_{Fit} also gives good confidences but its improvement against the WV strategy becomes much more difficult due to its fine-tuning. Otherwise C4.5 and Ripper are known not to be very good at estimating confidence degrees, whereas 3NN using the distance based confidence and after removing noisy instances is capable of providing useful degrees so as to take advantage of NMC.

6.4. NMC vs. state-of-the-art combinations (Comb and DCS)

In this last analysis, we wanted to test the NMC approach against state-of-the-art combinations for each base classifier and

Table 14
Friedman aligned-rank tests comparing the best combination model (Comb), dynamic OVO approach (DCS) and NMC with CS and IS using f_A in each base classifier with accuracy.

Methods	3NN	SVM_{Poly}	SVM_{Puk}	SVM_{Fit}	C4.5	Ripper
Comb	37.65 (0.98862)	42.00 (0.00189*)	42.41 (0.03965*)	32.20 (0.84271)	42.07 (0.01119*)	39.93 (0.25271)
DCS	33.74 (0.98862)	40.57 (0.00218*)	28.63	31.02	25.67	30.89
NMC with CS_IS-f_A	33.61	22.43	33.96 (0.36797)	41.78 (0.13785)	37.26 (0.05017+)	34.17 (0.57899)

A '+' means that there are statistical differences with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence)

the DCS approach. Notice that in this case, Comb in C4.5 and Ripper coincides with WV. In the case of the DCS, it is interesting to analyze how the static selection of NMC behaves compared with the dynamic alternative, which requires of all the OVO classifiers and also needs to compute the *k*NN of the instance to be classified in the original input space. In the case of NCM, the original input space is no longer used and *k*NN is only performed with respect to the selected instances.

Looking at Table 4, NMC is always superior to Comb in terms of global average accuracy (except for the case of SVM_{Fit}). With respect to DCS, differences are smaller, with the only exception of SVM_{Poly} in which NMC stands out. Logically, in terms of CS there is no doubt that NMC requires less classifiers, since it is the only one allowing for classifier pruning (in DCS they are dynamically used depending on the instance to be classified, but all of them are required). In order to shed light on the differences among these approaches, the Friedman aligned-rank test for each base classifier comparing these three methods are shown in Table 14.

Attending at these results, NMC is the best model in 3NN and SVM_{Poly} (in the latter case with significant differences with respect to the other two combinations). In the rest of the cases, DCS performs better in terms of ranks, but there are no statistical differences against NMC except for C4.5, whose confidence degrees seems not to be adequate for NMC. Hence, with NMC one is able to reduce the number of classifiers to near 50% in OVO without a significant loss of accuracy, and even increasing it when the base classifier provides good confidence estimates. In this cases, NMC can learn from both competent and non-competent classifiers the outputs that are expected for the instances of each class.

6.5. A deeper insight into NMC: discussion and lessons learned

With NMC we have presented a different view of the aggregation in OVO and the possibility of reducing classifiers. In this section, we first want to focus on an example of the results obtained, which are presented in the Appendix A. Consider Balance data-set in all the base classifiers, which is a 3-class problem, and therefore 3 classifiers are considered in OVO. We consider the methods in the previous section for the analysis (Comb, DCS and NMC with CS_IS using f_A):

- *3NN*. Classifier reduction barely achieves a 20% (averaged over 25 executions), which means that most of the times one classifier is removed, whereas in the rest of the cases all of them are considered. Accuracy rate is improved with respect to Comb and DCS.
- *SVM_{Poly}*. Around 50% of the classifiers can be removed in this case, which means that in several partitions only one classifier (which is trained for two classes) is able to classify the third one. Hence, there may be a relation between classes in such a way that the confidence degrees returned by SVM allow NMC to distinguish the third class; that is, the examples from the third class have a specific behavior with the SVM learned with the other two classes. Even with this level of pruning, overall results are improved with respect to DCS and Comb.
- *SVM_{Puk}*. In this case there is no reduction at all (20-30% can be achieved by forcing CS reduction, improving results in this case). The characteristics of the confidence degrees given by this configuration make it difficult to maintain accuracy while reducing classifiers and instances in training, as we have already explained.
- *SVM_{Fit}*. The behavior of this classifier is between that of the previous SVMs. Few classifiers are removed with f_A (6.66%), but almost 50% of them are removed when classifier reduction is forced achieving an even better accuracy than without doing so (only for 0.08%). The accuracy of DCS is improved in both cases, whereas Comb equals the accuracy of NMC with f_A.
- *C4.5*. This is an interesting case. There is a 66.7% of reduction, which means that only one classifier is used to classify all the three classes. However, this has nothing to do with the case of SVM_{Poly}. The problem here is that the original C4.5 with OVO is not capable of learning to classify one of the classes. In this case, our model learns that two classifiers are useless, and removes them. This is due to the class imbalance problem in Balance [63]. We are aware of this problem and we have already dealt with it in OVO [25,64], but we do not treat it specifically in this work in order to ease the comprehension of the proposed model (it is independent of the underlaying classifiers).
- *Ripper*. Surprisingly, in this classifier there is no way to reduce classifiers in any of the partitions (0% reduction). That is, the rules learned by Ripper for one pair of classes are not capable of helping us in classifying the remaining one. This behavior could be related with the inner features of this algorithm, which is mainly based in a "one-class" learning procedure.

From a more general viewpoint, we can summarize the lessons learned in the empirical study as follows:

a) WV is not capable of dealing with classifier reduction even after being adapted to handle the absence of classifiers in the aggregation. Otherwise, DTs are clearly benefited from CS, even in the case when it is forced. In fact, it can be considered to be a requisite in order to make them work properly.
b) In the case of NMC, forcing CS usually comes along with a decrease in accuracy, although a trade-off between reduction and performance could be sought depending on the application. In other respect, forcing IS is not as harmful as doing so with CS, as it could be expected. In fact, depending on the characteristics of the base classifier and its confidence degrees, forcing IS can be beneficial.
c) Among NMC models, all of them have their strengths and weaknesses. In first place, NMC without any selection (NoGA) offers very competitive results, but they are dependent on the base classifiers. In those cases where NoGA does not achieve the best result, applying IS allows one to remove noisy instances that are hindering the classification. However, none of these approaches allow for CS, which the CS_IS model does, being also capable of maintaining a competitive performance over all the base classifiers.
d) With respect to the models in which CS has been tested (WV, NMC and DTs), NMC can be considered to be the most suitable approach. It should be mentioned that removing classifiers (statically) has allowed us to outperform robust strategies as WV (even without considering reduction in this case).

e) NMC is competitive against the state-of-the-art aggregation strategies for OVO, while introducing the possibility of pruning classifiers. With respect to the dynamic approach, each one has its own advantages. Whereas DCS requires all the base classifiers and computing *k*NN in testing phase, NMC only stores a subset (near to 50%) of the classifiers and its aggregation consists of computing *k*NN but only over the selected score-matrices. In terms of accuracy, NMC is better suited for those classifiers providing good confidence degrees, whereas DCS performs better in the other cases. Anyway significant differences between these approaches are only found with SVM_{Poly} (in favor of NMC) and C4.5 (in favor of DCS).

Finally, looking at the data characteristics from Tables 2 and 3 and the complete results in the Appendix A, some straightforward conclusions can be drawn:

- The greater the number of classes in the data-set is, the greater the ensemble size is.
- The greater the ensemble size is, the greater classifier reduction achieved can be. This fact can be understood since having more classifiers implies more redundant information in their outputs, as the same instances are used in more classifiers.
- The greater the number of instances is, the greater the instance reduction achieved can be. Similarly, with more instances for the same number of classes, it becomes easier to discard not useful or redundant instances.

7. Concluding remarks and future research lines

In this paper, we have presented NMC model for OVO. This model is based on comparing the outputs (score-matrix) of the instance to be classified with those of the instances in the training set. Hence, the outputs of the classifiers are viewed as a transformation of the input space and considered for a new classification problem where *k*NN is applied. This novel view of the aggregation problem in OVO allows one to prune this type of classifier ensemble, which has not been previously considered.

With less classifiers, NMC is able to maintain classification accuracy and even outperform previous combination models if the confidence degrees of the base classifiers are adequate. In order to show the usefulness of the method, we have performed a thorough empirical analysis considering six base classifiers. We have shown that using different fitness functions in NMC classifier or instance reduction can be forced, depending on the user requirements for the tradeoff between accuracy and simplicity. Moreover, we have extended WV and DTs to allow for classifier pruning. However, even with this modification, WV is not able to manage the absence of classifiers properly, whereas DTs need it in order to achieve their maximum potential (but it cannot reach that of NMC).

This paper opens up new possibilities in terms of combining classifiers, both in decomposition strategies and classical ensembles. On account of the global analysis and the results analyzed for Balance dataset performed in the previous section, it is clear that the behavior of each base classifier is different even for the same data. Hence, our method does not only depend on the overlapping or relations between classes, but also in the way the classification boundaries are learned by the classifier. In this respect, several future lines arise:

- To study the information given by the selected classifiers and analyze whether it is related with the overlapping between classes or other complexity measures [65,66].
- To go a step further trying to use complexity measures (based on each classifier) to obtain which classifiers are required a priori (pre-pruning).
- To use the information about the classifiers selected in order to analyze which classes require a greater number of classifiers to be recognized. This knowledge could be coupled with the difficult classes problem in the OVO strategy studied in [56].
- To study the combination of DCS and NMC developing a double classifier selection scheme. It should be studied whether these models are diverse in terms of the classifiers selected in order to analyze the gain that can be obtained from their combination.

Apart from these future research lines focused on decomposition strategies and mainly on OVO, NMC model can be extended to any classifier ensemble, whose performance should be studied and compared against existing models in classical classifier ensembles. Furthermore, any other IS or FS algorithm can be considered instead of the well-known CHC algorithm. This work leaves open this possibility, since different synergies may be found using other models.

Acknowledgments

This work was partially supported by the Spanish Ministry of Education and Science under Projects TIN2011-28488, TIN2013-40765-P; TIN2014-57251-P; TIN2016-77356-P; the Andalusian Research Plan P10-TIC-6858, P11-TIC-7765; and both the University of Jaén and Caja Rural Provincial de Jaén under project UJA2014/06/15.

Appendix A. Detailed tables of results

In this appendix, we present the detailed results for all the methods tested in all the base classifiers and data-sets. These results are shown in Tables A.15 (3NN), A.16 (SVM_{Poly}), A.17 (SVM_{Puk}), A.18 (SVM_{Fit}), A.19 (C4.5) and A.20 (Ripper). The average accuracy rates in test, the instance and classifier reduction rates and the training times are presented in these tables. The best result within each base classifier and data-set is stressed in **bold-face**.

Table A.15
Average accuracy results in test, classifier and instance reduction rates and training times for 3NN classifier.

Accuracy			WV			NMC with $k=1$										DT		
				CS			CS		IS		CS_IS						CS	
Data-set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	**78.88**	76.96	75.83	75.12	74.88	77.16	74.81	74.45	76.68	75.85	76.70	75.90	76.82	76.87	75.91	71.21	73.39	74.10
Balance	87.20	87.20	87.20	87.19	87.19	90.72	90.40	90.40	92.28	92.22	92.19	**92.38**	92.25	92.15	92.22	86.88	87.84	87.84
Car	93.57	93.40	93.40	93.40	93.40	91.84	91.84	91.84	**96.02**	95.78	95.45	95.45	95.65	95.60	95.61	92.19	92.82	92.82
Cleveland	58.31	57.96	56.95	57.05	56.40	52.93	55.00	55.27	57.38	57.52	57.72	57.71	58.46	**58.53**	57.63	57.96	57.07	57.81
Dermatology	92.14	**95.49**	92.68	92.68	93.53	93.82	93.42	92.86	93.14	94.37	93.49	93.78	93.33	93.95	93.89	92.41	93.21	93.21
Ecoli	81.66	82.52	**82.52**	81.58	81.86	74.80	73.06	73.12	80.89	81.16	80.25	80.07	80.04	80.79	79.59	79.58	79.97	80.36
Glass	73.35	**74.27**	73.81	71.60	71.42	70.22	65.27	64.21	68.47	68.02	66.92	67.05	67.99	67.72	66.70	65.36	66.81	66.58
Hayes-roth	75.82	74.34	74.34	75.88	75.88	79.73	77.11	77.11	81.31	79.92	79.45	79.13	79.46	79.32	79.46	**81.92**	**81.92**	**81.92**
Iris	95.33	95.33	**95.33**	**95.33**	**95.33**	90.67	90.67	90.67	94.27	94.00	94.00	94.40	93.73	94.00	93.73	**95.33**	**95.33**	**95.33**
Lymphography	68.19	**79.55**	68.88	69.25	69.25	65.62	65.65	65.51	76.11	76.40	76.94	75.95	75.95	75.80	76.51	77.56	77.56	76.74
NewThyroid	96.28	96.28	96.28	96.28	96.28	92.09	92.09	92.09	96.37	96.74	96.47	96.47	96.74	96.74	96.74	96.28	**97.21**	**97.21**
Nursery	93.29	93.29	93.37	93.44	93.44	89.38	89.21	89.23	93.89	**94.05**	93.69	93.88	93.97	93.89	93.81	91.36	93.29	93.29
Page-blocks	95.63	95.46	95.27	95.30	95.19	94.93	95.20	95.09	95.49	**95.74**	95.41	95.45	95.49	95.49	95.52	80.72	95.12	95.16
Penbased	**97.00**	96.64	96.55	96.55	96.55	96.45	95.10	94.80	94.70	94.46	94.90	94.88	94.86	94.36	94.62	79.47	90.20	90.20
Satimage	87.58	87.73	87.73	87.73	87.73	86.18	86.47	86.44	88.18	88.27	88.08	88.36	88.17	88.39	**88.70**	84.49	87.96	87.87
Segment	96.58	**96.80**	96.71	96.71	96.71	94.72	95.00	95.08	96.56	96.53	96.72	96.70	96.74	96.72	96.57	93.20	96.38	96.44
Shuttle	**99.50**	99.40	99.40	99.40	99.40	99.22	99.37	99.37	99.43	99.39	99.43	99.42	99.43	99.43	99.42	97.84	99.34	99.36
Splice	93.41	93.72	94.04	94.04	94.04	**95.29**	**95.29**	**95.29**	94.98	94.35	94.55	94.99	94.72	94.60	94.41	94.04	94.04	94.04
Tae	44.25	44.25	44.25	39.09	39.09	55.59	**56.90**	56.21	40.94	41.84	42.26	43.96	45.21	43.03	42.18	37.09	36.44	36.44
Thyroid	94.72	94.16	94.72	94.72	94.72	18.32	18.32	18.32	94.50	94.61	94.97	**95.02**	94.83	95.00	94.91	91.39	93.19	93.19
Vowel	**97.78**	97.37	97.37	97.37	97.37	96.87	96.81	96.65	95.31	95.11	96.02	95.52	95.09	95.33	95.62	85.35	95.62	95.54
Wine	95.49	95.49	95.49	95.49	95.49	**99.43**	**99.43**	**99.43**	**99.43**	98.40	**99.43**	98.97	98.29	98.29	98.29	93.79	95.49	95.49
Zoo	89.90	91.86	90.74	91.13	91.33	92.64	93.11	90.79	92.54	92.01	92.68	92.28	92.87	92.66	92.94	93.64	**94.44**	92.79
Average	86.34	86.93	86.21	85.93	85.93	82.55	82.15	81.92	86.91	86.81	86.86	86.86	**86.96**	86.90	86.74	83.44	85.85	85.81
Classifier Reduction																		
Autos	–	–	–	1.33	4.27	–	46.40	49.07	–	–	55.47	**64.00**	58.40	61.33	60.00	–	39.47	39.73
Balance	–	–	–	**26.67**	**26.67**	–	20.00	**26.67**	–	–	14.67	17.33	10.67	12.00	17.33	–	13.33	13.33
Car	–	–	–	0.00	0.00	–	0.00	0.00	–	–	18.67	**23.33**	22.67	18.67	20.00	–	**23.33**	**23.33**
Cleveland	–	–	–	41.20	43.60	–	68.40	69.60	–	–	78.00	**81.60**	79.60	75.60	79.60	–	50.80	56.00
Dermatology	–	–	–	8.00	32.00	–	41.60	56.00	–	–	44.27	60.53	54.67	57.87	60.27	–	45.87	**61.33**
Ecoli	–	–	–	7.14	11.14	–	65.00	68.86	–	–	67.57	**72.86**	70.86	71.86	71.57	–	59.57	64.29
Glass	–	–	–	32.00	36.95	–	54.29	64.57	–	–	68.57	71.05	70.86	72.38	**74.48**	–	60.38	72.38
Hayes-roth	–	–	–	13.33	13.33	–	**26.67**	**26.67**	–	–	18.67	20.00	20.00	21.33	22.67	–	6.67	13.33
Iris	–	–	–	0.00	0.00	–	0.00	**33.33**	–	–	14.67	**33.33**	24.00	**33.33**	**33.33**	–	0.00	**33.33**
Lymphography	–	–	–	9.33	20.67	–	20.67	56.67	–	–	43.33	**59.33**	50.67	52.00	58.00	–	13.33	43.33
NewThyroid	–	–	–	0.00	0.00	–	0.00	26.67	–	–	18.67	**33.33**	**33.33**	**33.33**	**33.33**	–	20.00	20.00
Nursery	–	–	–	6.00	8.00	–	14.00	54.00	–	–	64.80	67.60	66.00	66.40	67.60	–	46.40	**70.00**
Page-blocks	–	–	–	27.60	30.80	–	48.40	52.80	–	–	54.40	62.80	55.60	58.80	60.40	–	62.00	**68.00**
Penbased	–	–	–	0.00	0.00	–	56.27	57.60	–	–	55.82	59.82	56.80	56.53	58.31	–	70.22	**70.40**
Satimage	–	–	–	0.19	0.00	–	52.95	59.43	–	–	71.62	**75.05**	74.86	73.71	74.48	–	59.24	65.52
Segment	–	–	–	0.00	0.57	–	40.76	45.90	–	–	55.81	**58.86**	56.00	56.57	56.95	–	51.43	54.29
Shuttle	–	–	–	0.00	24.00	–	39.20	58.00	–	–	44.80	**64.80**	56.00	62.40	**64.80**	–	51.20	58.00
Splice	–	–	–	**0.00**	**0.00**	–	**0.00**	**0.00**	–	–	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	–	**0.00**	**0.00**
Tae	–	–	–	**40.00**	**40.00**	–	13.33	20.00	–	–	34.67	36.00	38.67	**40.00**	38.67	–	33.33	33.33
Thyroid	–	–	–	0.00	0.00	–	13.33	13.33	–	–	25.33	33.33	29.33	33.33	33.33	–	**53.33**	**53.33**
Vowel	–	–	–	0.00	0.00	–	60.73	62.47	–	–	56.80	53.89	55.78	55.71	58.69	–	68.87	**70.11**
Wine	–	–	–	0.00	0.00	–	0.00	26.67	–	–	5.33	**33.33**	26.67	**33.33**	**33.33**	–	**33.33**	**33.33**
Zoo	–	–	–	7.81	24.95	–	51.81	70.48	–	–	51.43	71.05	61.14	70.86	71.81	–	8.57	**73.33**
Average	–	–	–	9.59	13.78	–	31.90	43.43	–	–	41.89	**50.14**	46.63	48.58	49.95	–	37.86	47.39
Instance Reduction																		
			WV			NMC with $k=1$										DT		
				CS			CS		IS		CS_IS						CS	
Data-set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	–	–	–	–	–	–	–	–	77.38	80.93	74.73	80.47	83.73	84.51	82.99	–	**95.49**	**95.49**
Balance	–	–	–	–	–	–	–	–	92.05	97.26	91.77	91.38	96.70	96.58	95.79	–	**99.40**	**99.40**
Car	–	–	–	–	–	–	–	–	92.57	95.26	90.53	89.81	95.22	95.46	94.74	–	**99.71**	**99.71**
Cleveland	–	–	–	–	–	–	–	–	92.76	93.57	92.80	92.66	94.73	94.34	94.34	–	**97.94**	**97.94**
Dermatology	–	–	–	–	–	–	–	–	69.65	94.82	74.21	82.98	95.36	95.32	95.23	–	**97.95**	**97.95**
Ecoli	–	–	–	–	–	–	–	–	91.11	92.93	92.09	92.16	93.38	93.44	92.95	–	**97.11**	**97.11**
Glass	–	–	–	–	–	–	–	–	84.75	86.87	83.76	87.48	89.75	89.47	89.50	–	**96.07**	**96.07**
Hayes-roth	–	–	–	–	–	–	–	–	81.24	90.22	80.24	80.23	90.76	91.21	90.53	–	**97.24**	**97.24**

(continued on next page)

Table A.15 (*continued*)

Accuracy																		
			WV			NMC with $k = 1$										DT		
				CS			CS		IS		CS_IS						CS	
Data-set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Iris	–	–	–	–	–	–	–	–	51.10	95.03	52.67	52.40	95.13	95.07	95.13	–	**97.56**	**97.56**
Lymphography	–	–	–	–	–	–	–	–	62.63	93.00	74.40	76.79	92.99	92.96	93.02	–	**96.73**	**96.73**
NewThyroid	–	–	–	–	–	–	–	–	59.35	96.42	63.58	66.07	96.53	96.51	96.56	–	**98.29**	**98.29**
Nursery	–	–	–	–	–	–	–	–	96.25	97.99	95.10	95.66	98.11	98.12	97.76	–	**99.52**	**99.52**
Page-blocks	–	–	–	–	–	–	–	–	81.88	97.09	81.81	83.19	97.57	97.56	97.52	–	**98.87**	**98.87**
Penbased	–	–	–	–	–	–	–	–	84.19	89.09	85.10	88.70	90.49	90.45	89.43	–	**98.88**	**98.88**
Satimage	–	–	–	–	–	–	–	–	94.32	95.56	93.62	94.86	96.22	96.02	96.15	–	**98.66**	**98.66**
Segment	–	–	–	–	–	–	–	–	95.98	97.69	96.37	96.88	97.93	98.25	97.76	–	**99.62**	**99.62**
Shuttle	–	–	–	–	–	–	–	–	55.84	99.40	64.25	70.58	99.39	99.39	99.39	–	**99.71**	**99.71**
Splice	–	–	–	–	–	–	–	–	72.92	97.34	80.23	78.21	97.27	97.27	97.27	–	**98.84**	**98.84**
Tae	–	–	–	–	–	–	–	–	72.71	75.06	72.42	71.78	75.57	76.97	76.86	–	**97.58**	**97.58**
Thyroid	–	–	–	–	–	–	–	–	82.75	98.81	84.30	86.25	98.72	98.69	98.67	–	**99.48**	**99.48**
Vowel	–	–	–	–	–	–	–	–	63.52	69.95	73.53	71.38	76.24	76.33	77.34	–	**98.63**	**98.63**
Wine	–	–	–	–	–	–	–	–	30.30	94.38	30.80	58.27	94.80	94.77	94.80	–	**97.94**	**97.94**
Zoo	–	–	–	–	–	–	–	–	46.48	83.17	59.83	71.54	82.66	82.66	82.71	–	**92.02**	**92.02**
Average	–	–	–	–	–	–	–	–	75.29	91.82	77.74	80.86	92.58	92.67	92.45	–	**97.97**	**97.97**
Training Times																		
Autos	–	–	–	0.24	0.22	–	2.44	2.42	3.50	4.45	3.47	3.38	4.22	4.01	4.39	–	0.22	**0.20**
Balance	–	–	–	0.10	0.04	–	1.13	1.01	16.25	14.33	17.67	18.39	15.92	17.31	18.09	–	0.13	**0.02**
Car	–	–	–	0.28	0.25	–	41.49	40.37	311.8	270.6	324.5	313.0	264.2	267.0	279.8	–	0.19	**0.15**
Cleveland	–	–	–	0.19	0.16	–	2.53	2.28	5.54	5.24	5.36	5.04	4.48	5.00	4.93	–	0.19	**0.07**
Dermatology	–	–	–	0.32	0.47	–	7.88	9.98	6.32	6.07	8.84	7.22	8.33	8.35	8.70	–	**0.24**	0.26
Ecoli	–	–	–	0.80	**0.75**	–	26.69	28.74	16.19	16.84	18.19	18.52	19.09	19.21	19.94	–	0.86	0.81
Glass	–	–	–	0.48	0.39	–	5.97	6.33	7.29	7.23	7.51	7.05	7.60	7.81	8.03	–	0.32	**0.25**
Hayes-roth	–	–	–	0.05	0.01	–	0.10	0.05	0.36	0.37	0.41	0.41	0.46	0.48	0.47	–	0.03	**0.01**
Iris	–	–	–	0.06	0.02	–	0.08	0.07	0.25	0.23	0.27	0.27	0.24	0.25	0.25	–	0.03	**0.01**
Lymphography	–	–	–	0.07	0.03	–	0.20	0.20	0.46	0.40	0.58	0.51	0.49	0.50	0.52	–	0.04	**0.02**
NewThyroid	–	–	–	0.06	0.02	–	0.16	0.14	0.79	0.65	0.78	0.74	0.45	0.45	0.46	–	0.04	**0.01**
Nursery	–	–	–	0.72	0.71	–	35.58	41.24	113.5	91.56	102.5	98.66	91.55	90.17	92.52	–	**0.20**	0.23
Page-blocks	–	–	–	0.26	0.18	–	8.39	7.96	13.55	9.93	15.16	12.69	8.52	8.16	8.89	–	0.15	**0.09**
Penbased	–	–	–	**6.03**	6.10	–	1141	1331	560.8	487.7	530.9	484.0	464.1	464.1	478.1	–	13.48	13.41
Satimage	–	–	–	0.97	**0.83**	–	56.52	54.47	45.80	43.29	44.21	39.09	43.39	46.21	44.69	–	0.85	1.01
Segment	–	–	–	4.68	4.87	–	994.3	1001	720.4	583.3	656.4	652.6	571.6	560.2	597.2	–	**3.95**	4.21
Shuttle	–	–	–	0.69	0.87	–	112.2	146.1	1145	97.40	865.1	696.8	116.0	123.6	128.5	–	0.44	**0.41**
Splice	–	–	–	0.06	0.02	–	0.35	0.32	2.21	2.19	2.41	2.36	2.44	2.50	2.27	–	0.04	**0.01**
Tae	–	–	–	0.05	0.01	–	0.11	0.08	2.28	2.34	2.21	2.12	2.40	2.33	2.43	–	0.03	**0.01**
Thyroid	–	–	–	0.08	0.05	–	1.21	1.15	14.10	6.93	13.20	12.81	7.04	7.23	6.91	–	0.05	**0.01**
Vowel	–	–	–	**6.46**	6.47	–	1489	1601	981.5	948.9	868.1	915.1	879.5	897.5	894.6	–	17.65	16.77
Wine	–	–	–	0.05	0.01	–	0.13	0.09	0.55	0.37	0.61	0.52	0.39	0.40	0.39	–	0.03	**0.01**
Zoo	–	–	–	0.22	0.17	–	0.82	1.25	0.46	0.61	0.99	1.12	1.18	1.38	1.41	–	**0.08**	0.10
Average	–	–	–	1.00	**0.98**	–	170.8	186.0	172.5	113.1	151.7	143.2	109.3	110.2	113.2	–	1.71	1.66

Table A.16
Average accuracy results in test, classifier and instance reduction rates and training times for SVM_{Poly} classifier.

Accuracy																		
			WV			NMC with $k = 1$										DT		
				CS			CS		IS		CS_IS						CS	
Data–set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	73.75	73.81	73.14	73.14	73.14	**75.35**	74.01	74.65	74.98	74.60	73.02	72.78	73.43	73.17	73.33	70.53	71.60	71.10
Balance	91.02	90.55	91.18	91.50	91.50	90.39	91.53	91.53	90.93	90.75	91.52	**92.13**	91.42	91.74	91.78	91.66	91.66	91.66
Car	93.58	93.58	93.69	93.69	93.69	94.85	94.82	94.79	**95.68**	95.60	95.58	95.56	95.67	95.65	95.59	91.78	92.22	92.21
Cleveland	58.97	59.31	59.32	58.30	58.30	52.49	52.08	51.88	58.36	59.39	59.17	**59.57**	58.03	58.62	58.83	57.95	57.79	58.75
Dermatology	94.71	94.99	94.71	94.71	94.42	**95.83**	**95.83**	94.64	**95.83**	95.25	**95.83**	94.53	95.27	94.20	94.02	95.55	95.55	94.92
Ecoli	79.37	79.63	78.78	78.72	78.72	76.82	75.73	75.90	77.40	77.81	**80.06**	80.02	80.03	79.21	78.84	71.51	75.54	75.23
Glass	62.14	63.14	63.14	61.98	62.91	**71.83**	66.73	65.63	67.34	66.59	67.41	67.46	66.52	67.22	67.98	55.22	60.88	61.28
Hayes-roth	54.45	53.74	52.31	52.31	52.31	**72.14**	70.60	70.60	66.71	67.63	67.96	67.81	67.07	67.53	67.64	56.81	53.08	52.92
Iris	95.33	95.33	95.33	95.33	95.33	**97.33**	96.00	96.40	96.67	96.53	96.00	96.27	95.87	96.00	95.87	95.33	95.33	95.33
Lymphography	82.48	82.48	82.48	82.48	82.48	81.85	80.05	79.87	80.60	79.94	80.12	81.47	80.52	80.53	81.08	**83.77**	83.37	83.50
NewThyroid	96.74	**97.21**	96.74	96.74	96.74	94.88	95.81	95.81	96.28	96.74	96.74	96.28	96.74	96.74	96.37	96.28	96.74	96.74
Nursery	92.13	92.13	92.13	92.37	92.21	89.59	89.52	89.45	92.28	92.17	92.37	**92.41**	92.41	92.35	92.35	89.60	90.07	90.07
Page-blocks	94.90	94.53	94.72	94.60	94.82	93.65	92.88	92.92	94.65	**94.94**	94.65	94.58	94.61	94.72	94.57	84.13	94.57	94.61
Penbased	95.92	96.01	95.83	95.83	95.83	**96.45**	96.44	96.15	95.48	95.32	95.79	95.30	95.44	95.61	95.63	80.11	88.78	88.45
Satimage	84.48	84.16	84.01	84.01	84.01	84.47	82.67	82.29	**84.82**	84.57	84.70	84.17	84.79	84.67	84.01	80.60	83.30	83.27
Segment	92.68	92.90	92.68	92.68	92.68	95.37	95.37	**95.41**	95.11	94.85	95.15	95.06	94.94	95.25	95.22	89.83	92.97	92.94
Shuttle	96.55	97.61	96.46	96.46	96.46	98.67	98.83	**98.83**	98.34	98.43	98.71	98.64	98.61	98.59	98.66	95.59	96.67	96.65
Splice	80.59	**80.91**	80.59	80.59	80.59	80.59	80.59	80.59	80.59	79.15	80.59	80.59	79.78	79.90	79.14	80.58	80.58	80.58
Tae	53.55	53.50	54.86	53.50	53.50	**63.51**	62.86	62.86	56.86	54.99	56.07	55.14	55.82	56.20	55.67	52.88	49.57	49.57
Thyroid	96.26	95.56	96.26	96.12	96.12	96.26	96.56	**96.84**	96.40	96.12	96.65	96.56	96.51	96.42	96.51	96.68	96.68	96.68

(*continued on next page*)

Table A.16 (*continued*)

Accuracy			WV			NMC with $k = 1$										DT		
				CS			CS		IS		CS_IS						CS	
Data–set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Vowel	71.41	71.82	71.11	71.11	71.11	91.41	**92.63**	92.55	86.95	86.85	87.49	87.72	87.03	87.21	87.76	60.51	69.70	69.98
Wine	**98.87**	**98.87**	98.87	98.87	98.87	98.33	98.87	98.87	98.33	98.87	98.87	98.87	98.55	98.55	98.55	98.87	98.87	98.87
Zoo	95.72	95.72	95.72	95.72	**95.73**	95.72	95.25	95.32	94.78	94.80	95.08	94.02	94.54	94.13	94.13	94.54	94.41	93.59
Average	84.16	84.24	84.09	83.95	83.98	**86.43**	85.90	85.82	85.89	85.73	86.07	85.95	85.81	85.84	85.81	81.32	83.04	83.00
Classifier Reduction																		
Autos	–	–	–	0.00	0.53	–	32.27	49.60	–	–	42.40	**50.93**	47.47	**50.93**	50.67	–	37.87	44.27
Balance	–	–	–	6.67	6.67	–	20.00	33.33	–	–	50.67	52.00	52.00	50.67	52.00	–	0.00	**66.67**
Car	–	–	–	0.00	0.00	–	18.00	20.00	–	–	26.00	**32.00**	28.67	28.67	30.67	–	23.33	23.33
Cleveland	–	–	–	23.60	30.00	–	65.20	68.00	–	–	68.80	72.00	71.60	71.60	**75.20**	–	56.80	59.60
Dermatology	–	–	–	0.00	37.33	–	8.00	62.67	–	–	9.60	65.60	47.47	59.73	**66.67**	–	17.87	57.33
Ecoli	–	–	–	1.43	1.14	–	62.71	68.57	–	–	77.71	80.00	77.86	78.86	**80.29**	–	72.00	75.14
Glass	–	–	–	16.00	28.00	–	56.38	64.38	–	–	70.29	**73.52**	70.48	71.05	72.00	–	59.81	69.14
Hayes-roth	–	–	–	0.00	6.67	–	40.00	53.33	–	–	50.67	58.67	56.00	60.00	**62.67**	–	26.67	60.00
Iris	–	–	–	0.00	0.00	–	40.00	**66.67**	–	–	37.33	64.00	36.00	44.00	64.00	–	0.00	33.33
Lymphography	–	–	–	3.33	13.33	–	24.00	50.00	–	–	36.67	**56.67**	52.00	**56.67**	**56.67**	–	7.33	46.67
NewThyroid	–	–	–	0.00	0.00	–	33.33	33.33	–	–	38.67	**46.67**	40.00	40.00	**46.67**	–	20.00	26.67
Nursery	–	–	–	8.00	12.00	–	20.40	60.00	–	–	66.40	**68.40**	67.20	68.00	**68.40**	–	30.40	44.00
Page-blocks	–	–	–	20.00	32.00	–	40.40	50.40	–	–	58.00	62.40	61.60	61.20	**63.20**	–	52.80	58.00
Penbased	–	–	–	0.00	0.00	–	56.80	58.76	–	–	57.51	61.96	58.76	59.38	62.22	–	71.73	**72.09**
Satimage	–	–	–	0.19	0.19	–	55.43	62.29	–	–	66.10	70.29	66.29	69.14	68.57	–	68.76	**72.57**
Segment	–	–	–	0.00	0.00	–	48.57	52.19	–	–	51.62	52.57	51.24	52.95	50.86	–	54.10	**55.24**
Shuttle	–	–	–	0.00	0.00	–	43.20	50.40	–	–	49.60	**58.80**	47.60	51.60	54.80	–	44.00	58.00
Splice	–	–	–	**0.00**	**0.00**	–	**0.00**	**0.00**	–	–	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	–	**0.00**	**0.00**
Tae	–	–	–	13.33	20.00	–	26.67	**40.00**	–	–	29.33	36.00	28.00	28.00	34.67	–	33.33	33.33
Thyroid	–	–	–	20.00	20.00	–	26.67	**33.33**	–	–	30.67	**33.33**	30.67	32.00	**33.33**	–	6.67	20.00
Vowel	–	–	–	0.00	0.00	–	61.82	63.20	–	–	44.44	42.91	48.44	44.80	40.44	–	**77.38**	76.95
Wine	–	–	–	0.00	0.00	–	6.67	**33.33**	–	–	6.67	**33.33**	26.67	**33.33**	**33.33**	–	0.00	26.67
Zoo	–	–	–	0.00	20.19	–	44.57	**80.76**	–	–	49.90	74.29	60.19	70.67	74.10	–	23.05	71.43
Average	–	–	–	4.89	9.92	–	36.13	50.20	–	–	44.31	**54.19**	48.96	51.45	53.97	–	34.08	50.02
Instance Reduction																		
			WV			NMC with $k = 1$										DT		
				CS			CS		IS		CS_IS						CS	
Data-set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	–	–	–	–	–	–	–	–	66.91	84.79	74.02	78.37	86.96	88.38	87.09	–	**95.49**	**95.49**
Balance	–	–	–	–	–	–	–	–	91.75	94.86	92.11	92.05	95.63	95.51	95.45	–	**99.40**	**99.40**
Car	–	–	–	–	–	–	–	–	93.78	97.95	93.11	92.77	97.84	97.77	97.67	–	**99.71**	**99.71**
Cleveland	–	–	–	–	–	–	–	–	92.79	93.99	93.35	93.10	94.24	94.09	93.52	–	**97.94**	**97.94**
Dermatology	–	–	–	–	–	–	–	–	0.00	95.81	12.99	74.98	95.81	95.81	95.71	–	**97.95**	**97.95**
Ecoli	–	–	–	–	–	–	–	–	88.03	89.84	90.81	91.98	93.09	92.45	93.04	–	**97.11**	**97.11**
Glass	–	–	–	–	–	–	–	–	83.97	86.49	86.11	85.36	86.90	86.74	87.02	–	**96.07**	**96.07**
Hayes-roth	–	–	–	–	–	–	–	–	77.22	89.39	77.52	79.96	89.44	90.38	89.96	–	**97.24**	**97.24**
Iris	–	–	–	–	–	–	–	–	60.70	94.37	57.47	64.27	94.20	94.10	93.43	–	**97.56**	**97.56**
Lymphography	–	–	–	–	–	–	–	–	65.26	92.91	65.43	69.34	93.42	93.42	93.42	–	**96.73**	**96.73**
NewThyroid	–	–	–	–	–	–	–	–	53.77	96.40	61.14	62.65	96.05	96.05	95.93	–	**98.29**	**98.29**
Nursery	–	–	–	–	–	–	–	–	95.92	98.17	96.04	95.49	98.36	98.17	98.02	–	**99.52**	**99.52**
Page-blocks	–	–	–	–	–	–	–	–	85.74	97.07	87.17	89.86	97.50	97.38	97.45	–	**98.87**	**98.87**
Penbased	–	–	–	–	–	–	–	–	85.07	88.68	84.50	87.69	90.13	90.87	90.62	–	**98.88**	**98.88**
Satimage	–	–	–	–	–	–	–	–	92.07	94.48	92.55	93.74	95.09	94.84	94.93	–	**98.66**	**98.66**
Segment	–	–	–	–	–	–	–	–	89.98	93.16	88.24	90.71	92.47	91.90	90.22	–	**99.62**	**99.62**
Shuttle	–	–	–	–	–	–	–	–	84.68	97.63	81.28	85.00	96.30	97.02	96.72	–	**99.71**	**99.71**
Splice	–	–	–	–	–	–	–	–	0.00	97.65	0.00	0.00	97.65	97.65	97.65	–	**98.84**	**98.84**
Tae	–	–	–	–	–	–	–	–	73.40	79.17	72.13	72.98	78.83	78.57	79.49	–	**97.58**	**97.58**
Thyroid	–	–	–	–	–	–	–	–	66.08	98.83	73.45	74.62	98.69	98.70	98.70	–	**99.48**	**99.48**
Vowel	–	–	–	–	–	–	–	–	55.27	56.81	55.20	51.90	59.06	57.83	52.19	–	**98.63**	**98.63**
Wine	–	–	–	–	–	–	–	–	0.00	95.79	10.38	48.91	95.79	95.79	95.79	–	**97.94**	**97.94**
Zoo	–	–	–	–	–	–	–	–	64.64	83.02	54.02	69.95	82.66	82.61	81.76	–	**92.02**	**92.02**
Average	–	–	–	–	–	–	–	–	68.13	91.18	69.52	75.90	91.57	91.57	91.12	–	**97.97**	**97.97**
Training Times																		
Autos	–	–	–	0.23	0.17	–	2.03	2.06	2.84	3.75	2.95	2.83	3.41	3.42	3.80	–	0.24	**0.11**
Balance	–	–	–	0.08	0.04	–	1.12	1.19	16.91	15.27	18.24	18.64	15.49	16.36	16.90	–	0.18	**0.01**
Car	–	–	–	0.27	0.23	–	63.77	68.90	338.6	277.8	345.1	325.8	278.1	286.2	288.2	–	0.21	**0.17**
Cleveland	–	–	–	0.17	0.10	–	3.30	2.87	4.88	4.84	5.64	5.93	5.72	5.64	6.33	–	0.18	**0.07**
Dermatology	–	–	–	0.27	0.29	–	4.60	8.55	14.11	3.67	11.51	7.17	5.76	6.86	9.15	–	**0.21**	0.24
Ecoli	–	–	–	0.72	0.63	–	26.38	26.93	21.76	22.95	19.86	19.28	20.85	21.42	21.15	–	0.58	**0.51**
Glass	–	–	–	0.48	0.40	–	5.94	6.59	9.09	8.77	9.14	8.84	10.12	10.07	10.32	–	0.27	**0.23**
Hayes-roth	–	–	–	0.05	0.01	–	0.09	0.05	0.38	0.47	0.36	0.33	0.50	0.47	0.45	–	0.02	**0.01**
Iris	–	–	–	0.04	0.02	–	0.09	0.07	0.30	0.25	0.37	0.39	0.29	0.31	0.33	–	0.03	**0.01**
Lymphography	–	–	–	0.08	0.04	–	0.21	0.23	0.48	0.39	0.56	0.50	0.37	0.37	0.40	–	0.05	**0.02**

(*continued on next page*)

Table A.16 (*continued*)

Accuracy			WV			NMC with $k=1$										DT		
				CS			CS		IS		CS_IS						CS	
Data–set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
NewThyroid	–	–	–	0.10	0.02	–	0.16	0.12	0.73	0.56	0.82	0.80	0.61	0.62	0.62	–	0.09	**0.01**
Nursery	–	–	–	0.87	0.87	–	30.38	40.14	104.4	91.84	108.0	94.22	89.13	86.05	87.33	–	**0.36**	0.37
Page-blocks	–	–	–	0.23	0.21	–	8.92	10.53	16.01	10.04	15.44	13.63	10.14	10.06	11.18	–	0.18	**0.13**
Penbased	–	–	–	**5.21**	5.34	–	1013	1216	569.6	506.5	533.6	497.2	492.9	471.2	479.8	–	12.05	12.54
Satimage	–	–	–	0.92	0.81	–	55.19	57.86	55.86	54.02	58.19	55.34	56.35	57.23	58.49	–	0.79	**0.71**
Segment	–	–	–	**3.97**	3.99	–	1399	1425	2288	1972	2007	1845	1760	1677	1827	–	4.32	4.36
Shuttle	–	–	–	0.84	0.81	–	129.6	146.3	912.0	524.2	814.9	718.4	517.2	500.4	556.0	–	**0.46**	0.51
Splice	–	–	–	0.06	0.02	–	0.38	0.35	4.74	0.85	4.01	4.02	1.04	1.04	1.05	–	0.04	**0.01**
Tae	–	–	–	0.05	0.01	–	0.10	0.07	1.68	1.72	1.82	1.78	2.03	2.22	2.17	–	0.03	**0.01**
Thyroid	–	–	–	0.07	0.04	–	1.49	1.60	24.45	6.52	23.37	21.60	6.96	7.36	7.25	–	0.10	**0.02**
Vowel	–	–	–	4.50	**4.27**	–	1766	1747	1475	1455	1348	1344	1338	1322	1369	–	17.71	17.98
Wine	–	–	–	0.05	0.01	–	0.13	0.10	0.81	0.28	0.71	0.49	0.31	0.30	0.32	–	0.03	**0.01**
Zoo	–	–	–	0.27	0.18	–	0.66	1.18	0.60	0.77	1.12	1.50	1.38	1.64	1.90	–	**0.12**	0.13
Average	–	–	–	0.85	**0.80**	–	196.2	207.1	254.9	215.8	231.8	216.9	200.7	195.1	206.9	–	1.66	1.66

Table A.17
Average accuracy results in test, classifier and instance reduction rates and training times for SVM_{Puk} classifier.

Accuracy			WV			NMC with $k=1$										DT		
				CS			CS		IS		CS_IS						CS	
Data-set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	69.02	70.27	69.61	69.61	59.61	71.53	71.54	70.81	**71.77**	70.35	**71.77**	70.72	70.05	71.61	70.23	67.04	70.12	70.51
Balance	89.58	88.46	89.58	89.58	**91.48**	89.58	89.58	89.58	89.58	91.31	89.58	89.99	90.53	90.63	91.14	90.37	90.37	90.37
Car	64.99	**84.84**	65.39	65.39	65.39	72.92	72.92	76.42	72.92	73.46	72.92	78.12	72.05	73.74	77.05	70.95	70.95	71.35
Cleveland	47.53	47.87	47.87	47.87	47.53	48.20	48.41	48.54	48.21	47.95	48.88	48.61	47.67	48.61	49.01	**51.90**	48.75	48.95
Dermatology	**97.20**	**97.20**	97.20	97.20	95.19	97.19	97.13	94.90	97.19	96.57	97.13	94.91	96.80	96.13	94.84	96.62	96.74	96.41
Ecoli	77.11	77.11	77.11	77.11	77.11	76.85	77.33	77.21	78.19	78.35	78.71	78.88	78.41	78.66	78.46	**81.38**	78.52	77.77
Glass	73.72	74.15	74.62	74.62	**74.72**	72.78	70.12	69.95	73.19	72.68	73.63	73.90	72.67	72.71	72.20	70.59	71.54	72.66
Hayes-roth	80.44	80.44	80.44	80.44	79.67	82.80	82.80	**82.95**	79.70	81.04	80.55	81.00	80.73	81.04	81.45	81.21	81.21	81.21
Iris	**95.33**	**95.33**	95.33	95.33	95.33	94.67	94.67	94.67	94.67	94.93	94.67	94.93	95.20	94.80	95.07	95.33	95.33	95.33
Lymphography	81.87	81.87	**81.87**	**81.87**	72.93	**81.87**	**81.87**	63.42	**81.87**	**81.87**	**81.87**	63.80	79.60	72.07	66.96	**81.87**	**81.87**	74.98
NewThyroid	**95.81**	**95.81**	95.81	95.81	95.81	95.81	95.81	95.53	95.81	95.72	95.81	95.81	95.81	95.81	95.81	95.81	95.81	95.81
Nursery	80.33	**89.05**	80.94	80.94	78.42	85.57	85.57	87.84	85.57	84.09	85.57	87.89	85.18	85.74	87.23	83.49	83.49	83.98
Page-blocks	94.58	94.76	94.76	94.76	94.75	94.04	93.82	93.67	94.83	94.45	94.53	94.68	94.68	94.68	94.68	93.66	94.52	**94.86**
Penbased	97.55	**97.64**	97.55	97.55	97.55	97.64	96.84	96.70	96.46	95.89	96.72	96.46	96.64	96.59	96.35	80.65	95.21	95.50
Satimage	84.77	85.70	85.38	85.38	85.26	85.55	85.02	84.71	**85.96**	85.24	85.30	85.33	85.43	85.34	85.27	85.73	85.40	85.24
Segment	97.23	**97.36**	97.32	97.32	97.32	97.23	97.23	97.16	97.29	97.31	97.26	97.17	97.14	97.15	97.16	95.33	97.17	97.15
Shuttle	99.59	**99.63**	99.59	99.59	99.59	99.59	99.50	99.47	99.63	99.60	99.50	99.51	99.52	99.50	99.50	99.59	99.59	99.61
Splice	65.57	**75.87**	63.38	63.38	59.53	64.31	64.31	44.26	64.31	65.70	64.31	30.88	55.88	56.28	52.94	65.89	65.89	62.30
Tae	52.81	51.50	52.79	52.79	52.79	**64.15**	62.55	62.42	56.25	56.76	58.26	57.42	56.46	57.67	56.89	52.15	52.15	52.15
Thyroid	94.31	93.89	**94.31**	**94.31**	**94.31**	93.89	93.89	93.61	94.02	93.44	93.86	93.69	93.66	93.63	93.60	94.02	94.02	94.30
Vowel	**99.70**	**99.70**	99.70	99.70	99.70	98.99	99.03	98.73	97.66	97.27	98.28	98.16	97.96	97.92	98.04	81.62	96.14	96.04
Wine	97.19	97.19	97.19	97.19	**97.78**	97.19	97.19	89.20	97.19	96.95	97.19	89.20	97.08	96.16	89.77	97.19	97.19	97.19
Zoo	78.06	84.13	82.45	82.45	84.26	87.85	87.85	87.28	87.85	87.85	87.85	89.44	87.97	89.42	**90.41**	87.85	87.85	89.64
Average	83.23	**85.21**	83.49	83.49	82.44	84.79	84.56	82.57	84.35	84.30	84.53	82.20	83.79	83.73	83.22	82.62	83.91	83.62
Classifier Reduction																		
Autos	–	–	–	0.00	50.67	–	23.20	52.00	–	–	26.93	**54.40**	50.67	52.00	53.87	–	40.27	50.67
Balance	–	–	–	0.00	13.33	–	0.00	6.67	–	–	0.00	20.00	0.00	21.33	**32.00**	–	0.00	0.00
Car	–	–	–	0.00	0.00	–	0.00	**63.33**	–	–	0.00	58.67	14.00	24.67	60.00	–	0.00	16.67
Cleveland	–	–	–	0.00	12.00	–	12.00	18.80	–	–	17.20	**28.80**	22.40	26.00	27.60	–	18.40	20.00
Dermatology	–	–	–	0.00	66.67	–	9.60	**79.73**	–	–	8.00	78.40	65.60	73.07	77.60	–	39.47	73.33
Ecoli	–	–	–	0.00	0.00	–	56.43	63.14	–	–	62.43	**72.71**	70.29	70.71	71.71	–	60.57	69.29
Glass	–	–	–	0.00	4.57	–	46.67	59.05	–	–	56.38	63.62	60.57	63.81	62.67	–	55.05	**64.76**
Hayes-roth	–	–	–	0.00	6.67	–	0.00	33.33	–	–	4.00	32.00	2.67	6.67	**34.67**	–	0.00	0.00
Iris	–	–	–	0.00	0.00	–	0.00	**33.33**	–	–	0.00	**33.33**	28.00	**33.33**	**33.33**	–	0.00	**33.33**
Lymphography	–	–	–	0.00	50.00	–	0.00	70.00	–	–	0.00	**73.33**	56.00	71.33	**73.33**	–	0.00	63.33
NewThyroid	–	–	–	0.00	0.00	–	0.00	**33.33**	–	–	0.00	**33.33**	30.67	**33.33**	**33.33**	–	0.00	**33.33**
Nursery	–	–	–	0.00	4.00	–	0.00	**70.00**	–	–	0.00	**70.00**	61.20	**70.00**	**70.00**	–	0.00	60.00
Page-blocks	–	–	–	0.00	32.00	–	23.20	**64.00**	–	–	37.20	55.60	49.60	52.80	57.20	–	35.60	54.00
Penbased	–	–	–	0.00	0.00	–	55.20	58.31	–	–	56.53	59.29	58.76	58.04	60.18	–	67.56	**68.53**
Satimage	–	–	–	0.00	0.57	–	50.48	60.38	–	–	51.81	**63.43**	60.95	59.62	63.24	–	54.48	59.62
Segment	–	–	–	0.00	0.76	–	15.05	56.19	–	–	33.90	**60.19**	56.57	56.76	59.81	–	51.24	54.48
Shuttle	–	–	–	0.00	20.00	–	34.40	**56.00**	–	–	44.00	53.60	47.60	48.00	53.60	–	19.20	50.00
Splice	–	–	–	0.00	33.33	–	0.00	**66.67**	–	–	0.00	**66.67**	52.00	**66.67**	**66.67**	–	0.00	60.00
Tae	–	–	–	0.00	0.00	–	26.67	**33.33**	–	–	1.33	2.67	0.00	0.00	8.00	–	0.00	0.00
Thyroid	–	–	–	0.00	0.00	–	0.00	**33.33**	–	–	13.33	**33.33**	**33.33**	**33.33**	**33.33**	–	6.67	20.00
Vowel	–	–	–	0.00	0.00	–	61.24	70.69	–	–	60.36	66.40	63.78	62.40	62.76	–	71.27	**71.49**

(*continued on next page*)

Table A.17 (*continued*)

Accuracy			WV			NMC with $k=1$										DT		
				CS			CS		IS		CS_IS						CS	
Data-set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Wine	–	–	–	0.00	13.33	–	0.00	**53.33**	–	–	0.00	**53.33**	29.33	36.00	52.00	–	0.00	33.33
Zoo	–	–	–	0.00	71.43	–	0.00	**85.14**	–	–	0.00	82.48	70.29	80.57	81.52	–	0.00	82.67
Average	–	–	–	0.00	16.49	–	18.01	53.05	–	–	20.58	52.85	42.79	47.85	**53.41**	–	22.60	45.17
Instance Reduction																		
			WV			NMC with $k=1$										DT		
				CS			CS		IS		CS_IS						CS	
Data–set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	–	–	–	–	–	–	–	–	37.43	88.30	45.28	75.12	90.56	90.56	90.44	–	**95.49**	**95.49**
Balance	–	–	–	–	–	–	–	–	0.00	98.80	0.00	34.33	98.80	98.80	98.80	–	**99.40**	**99.40**
Car	–	–	–	–	–	–	–	–	0.00	99.42	0.00	54.91	99.42	99.42	99.33	–	**99.71**	**99.71**
Cleveland	–	–	–	–	–	–	–	–	50.71	93.96	58.67	74.98	95.17	95.51	95.56	–	**97.94**	**97.94**
Dermatology	–	–	–	–	–	–	–	–	9.68	95.81	10.25	73.18	95.81	95.81	95.39	–	**97.95**	**97.95**
Ecoli	–	–	–	–	–	–	–	–	75.94	92.56	78.64	89.56	94.54	94.22	94.48	–	**97.11**	**97.11**
Glass	–	–	–	–	–	–	–	–	80.14	90.77	81.43	85.73	92.10	92.19	91.70	–	**96.07**	**96.07**
Hayes-roth	–	–	–	–	–	–	–	–	61.48	94.32	68.00	73.32	94.32	94.32	92.80	–	**97.24**	**97.24**
Iris	–	–	–	–	–	–	–	–	0.00	95.33	0.00	47.00	95.33	95.33	95.33	–	**97.56**	**97.56**
Lymphography	–	–	–	–	–	–	–	–	0.00	93.93	0.00	60.55	93.93	93.93	93.93	–	**96.73**	**96.73**
NewThyroid	–	–	–	–	–	–	–	–	0.00	96.51	0.00	47.58	96.51	96.51	96.51	–	**98.29**	**98.29**
Nursery	–	–	–	–	–	–	–	–	0.00	99.23	0.00	61.23	99.23	99.23	99.23	–	**99.52**	**99.52**
Page-blocks	–	–	–	–	–	–	–	–	73.48	97.46	63.30	72.02	97.66	97.66	97.57	–	**98.87**	**98.87**
Penbased	–	–	–	–	–	–	–	–	76.46	85.16	81.02	84.93	91.51	91.06	90.27	–	**98.88**	**98.88**
Satimage	–	–	–	–	–	–	–	–	85.21	95.92	78.64	87.49	97.15	97.08	97.02	–	**98.66**	**98.66**
Segment	–	–	–	–	–	–	–	–	57.60	98.30	55.84	74.73	99.05	99.06	98.91	–	**99.62**	**99.62**
Shuttle	–	–	–	–	–	–	–	–	13.06	99.40	67.13	65.49	99.37	99.36	99.36	–	**99.71**	**99.71**
Splice	–	–	–	–	–	–	–	–	0.00	97.65	0.00	49.72	97.65	97.65	97.65	–	**98.84**	**98.84**
Tae	–	–	–	–	–	–	–	–	78.55	91.75	80.80	83.47	91.62	92.31	90.69	–	**97.58**	**97.58**
Thyroid	–	–	–	–	–	–	–	–	39.53	98.96	42.54	54.86	98.96	98.96	98.96	–	**99.48**	**99.48**
Vowel	–	–	–	–	–	–	–	–	61.07	74.20	71.92	79.00	83.77	83.83	82.55	–	**98.63**	**98.63**
Wine	–	–	–	–	–	–	–	–	0.00	95.79	0.00	50.83	95.79	95.79	95.79	–	**97.94**	**97.94**
Zoo	–	–	–	–	–	–	–	–	0.00	82.66	0.00	66.72	82.66	82.66	82.55	–	**92.02**	**92.02**
Average	–	–	–	–	–	–	–	–	34.80	93.75	38.41	67.25	94.82	94.84	94.56	–	**97.97**	**97.97**
Training Times																		
Autos	–	–	–	0.23	0.18	–	1.70	2.00	1.38	1.76	1.83	1.65	1.79	1.82	2.04	–	0.21	**0.11**
Balance	–	–	–	0.13	0.04	–	1.34	1.29	36.35	3.00	30.34	24.87	3.62	4.10	4.39	–	0.22	**0.02**
Car	–	–	–	0.27	0.23	–	23.24	31.45	1260	40.49	922.1	488.1	61.69	66.35	66.87	–	0.19	**0.18**
Cleveland	–	–	–	0.21	0.19	–	4.99	4.98	5.06	5.39	5.69	5.47	5.75	5.65	5.99	–	0.27	**0.10**
Dermatology	–	–	–	0.34	0.21	–	2.99	7.05	12.81	3.18	12.20	7.66	4.28	4.60	8.01	–	0.21	**0.15**
Ecoli	–	–	–	0.81	0.67	–	23.68	26.27	17.33	17.95	15.49	13.64	15.93	15.72	17.18	–	0.69	**0.65**
Glass	–	–	–	0.46	0.36	–	6.11	6.40	3.76	4.73	5.07	5.00	5.85	5.71	6.23	–	0.28	**0.20**
Hayes-roth	–	–	–	0.05	0.01	–	0.10	0.07	0.18	0.19	0.21	0.21	0.20	0.21	0.29	–	0.03	**0.01**
Iris	–	–	–	0.05	0.02	–	0.11	0.09	0.50	0.20	0.47	0.30	0.23	0.23	0.25	–	0.03	**0.01**
Lymphography	–	–	–	0.08	0.03	–	0.14	0.19	0.60	0.20	0.68	0.47	0.29	0.33	0.38	–	0.04	**0.02**
NewThyroid	–	–	–	0.10	0.02	–	0.20	0.15	1.53	0.42	1.40	0.91	0.45	0.46	0.48	–	0.09	**0.01**
Nursery	–	–	–	0.71	0.70	–	21.18	42.71	597.0	19.15	445.0	224.2	26.80	27.83	29.94	–	**0.22**	0.26
Page-blocks	–	–	–	0.23	0.17	–	6.52	8.10	19.23	7.25	22.42	17.58	7.22	7.09	8.06	–	0.17	**0.11**
Penbased	–	–	–	4.67	**4.37**	–	956.6	1056	732.2	617.6	531.3	492.9	425.1	422.1	452.3	–	15.38	14.33
Satimage	–	–	–	0.96	0.96	–	49.55	53.31	58.07	52.03	53.76	43.22	43.98	44.66	47.67	–	**0.87**	0.91
Segment	–	–	–	4.31	4.41	–	613.3	877.5	2123	542.7	1942	1106	388.7	398.4	418.2	–	**3.36**	3.54
Shuttle	–	–	–	0.79	1.07	–	126.3	160.2	1834	117.6	985.8	909.0	157.9	152.0	169.8	–	**0.54**	0.56
Splice	–	–	–	0.07	0.03	–	0.20	0.27	5.07	0.84	4.94	2.84	0.92	0.96	0.99	–	0.04	**0.01**
Tae	–	–	–	0.05	0.01	–	0.10	0.08	0.52	0.69	0.53	0.62	0.80	0.74	0.81	–	0.03	**0.01**
Thyroid	–	–	–	0.09	0.05	–	1.54	1.44	36.56	5.21	34.23	28.40	4.88	4.88	4.88	–	0.07	**0.02**
Vowel	–	–	–	**5.17**	5.43	–	1041	1112	1012	942.5	872.7	800.2	769.9	775.5	798.9	–	18.86	19.77
Wine	–	–	–	0.05	0.02	–	0.13	0.09	0.83	0.27	0.79	0.53	0.30	0.31	0.30	–	0.03	**0.01**
Zoo	–	–	–	0.23	0.10	–	0.26	0.92	0.46	0.64	0.60	1.00	1.02	1.27	1.35	–	0.08	**0.07**
Average	–	–	–	0.87	**0.84**	–	125.3	147.5	337.3	103.7	256.1	181.5	83.81	84.39	88.93	–	1.82	1.78

Table A.18
Average accuracy results in test, classifier and instance reduction rates and training times for SVM_{Fit} classifier.

Accuracy																		
			WV			NMC with $k=1$										DT		
				CS			CS		IS		CS_IS						CS	
Data–set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	81.31	81.31	81.31	81.31	81.31	81.24	80.91	81.30	81.52	81.90	80.39	79.05	79.01	78.76	79.90	80.09	**82.25**	81.18
Balance	98.88	97.60	98.88	98.88	98.88	98.88	98.88	**99.36**	98.91	99.20	98.82	99.00	99.01	99.13	99.23	98.88	98.88	98.88
Car	99.48	99.48	99.54	99.54	99.54	99.59	99.59	99.51	99.59	99.24	99.59	**99.62**	99.28	98.90	97.92	99.25	99.32	99.37
Cleveland	58.32	59.67	59.67	59.33	**59.80**	53.84	55.33	55.19	59.28	58.79	57.51	56.51	57.39	57.83	56.70	58.27	57.60	58.40
Dermatology	96.91	97.20	96.91	96.80	95.95	97.18	**97.41**	95.48	96.56	96.29	96.63	95.16	96.13	96.07	95.50	95.49	96.68	96.06
Ecoli	**84.06**	83.47	83.47	83.47	83.47	78.89	77.31	76.82	81.91	81.14	83.22	82.69	83.09	83.09	83.48	77.16	82.53	82.70
Glass	66.89	68.23	68.70	68.70	68.70	**70.97**	70.15	69.96	68.90	68.45	68.93	68.42	67.61	67.75	67.10	65.58	66.56	67.20
Hayes-roth	79.67	78.90	78.90	78.29	79.05	82.03	82.80	**83.23**	78.34	79.53	79.81	81.16	80.11	80.71	81.29	80.44	80.44	80.13
Iris	97.33	97.33	97.33	97.33	97.33	98.00	97.33	97.33	98.00	98.00	**98.13**	97.60	98.00	98.00	97.87	97.33	97.33	97.33
Lymphography	82.43	82.43	82.43	82.43	83.05	81.03	79.81	79.01	81.69	81.24	81.79	81.50	81.10	81.76	81.62	**83.72**	**83.72**	83.45
NewThyroid	95.81	95.81	95.81	95.81	95.81	93.95	95.81	96.19	95.91	95.26	95.35	95.53	95.81	95.63	94.70	**96.28**	**96.28**	95.81
Nursery	**98.61**	**98.61**	**98.61**	**98.61**	**98.61**	**98.61**	**98.61**	97.92	**98.61**	98.20	**98.61**	98.46	**98.61**	98.58	**98.61**	95.10	95.10	94.33
Page-blocks	94.92	94.92	95.09	95.09	94.91	94.40	93.60	93.63	**95.29**	95.01	94.89	94.70	94.78	94.82	94.85	88.86	95.27	95.24
Penbased	97.55	**97.64**	97.55	97.55	97.55	97.46	96.71	96.77	96.21	96.20	96.44	96.15	96.39	96.28	96.06	81.83	94.64	94.55
Satimage	89.92	**90.23**	**90.23**	**90.23**	**90.23**	88.81	87.94	87.85	89.48	89.14	89.36	89.45	88.96	89.36	89.21	87.28	89.33	89.40
Segment	97.01	97.01	97.01	97.01	97.04	97.06	96.96	96.84	**97.18**	97.15	97.04	97.00	97.08	97.00	96.96	95.15	96.93	96.95
Shuttle	99.72	**99.72**	99.68	99.68	99.72	99.59	99.50	99.37	99.59	99.64	99.61	99.62	99.56	99.58	99.60	99.63	99.68	99.67
Splice	90.96	**92.19**	91.58	91.58	91.58	89.03	89.03	89.03	90.20	90.31	90.44	90.57	90.49	90.37	90.49	91.88	91.88	91.88
Tae	60.09	60.11	61.42	61.42	61.42	**66.75**	64.17	64.82	59.61	59.77	58.72	58.59	59.25	58.60	59.00	60.75	60.75	60.75
Thyroid	96.54	95.84	96.67	96.67	96.67	96.95	97.51	**97.65**	97.17	97.34	97.04	97.04	97.20	97.18	97.18	96.68	96.81	96.67
Vowel	**99.70**	**99.70**	99.70	99.70	99.70	98.99	98.89	98.73	97.76	97.49	98.18	98.14	97.98	98.12	98.26	81.72	95.54	96.02
Wine	**98.33**	**98.33**	**98.33**	**98.33**	**98.33**	**98.33**	**98.33**	**98.33**	**98.33**	**98.33**	**98.33**	**98.33**	98.22	**98.33**	**98.33**	**98.33**	**98.33**	**98.33**
Zoo	94.72	94.72	94.72	94.72	**95.72**	**95.72**	94.45	91.62	95.01	95.71	95.12	94.64	94.23	93.16	94.19	94.54	94.54	93.52
Average	89.53	89.59	89.72	89.67	**89.76**	89.45	89.18	88.95	89.35	89.27	89.30	89.08	89.10	89.09	89.04	87.14	89.15	89.04
Classifier Reduction																		
Autos	–	–	–	0.00	0.00	–	36.80	45.33	–	–	29.60	**52.27**	45.33	49.87	49.33	–	36.80	42.67
Balance	–	–	–	0.00	0.00	–	0.00	**46.67**	–	–	6.67	**46.67**	38.67	41.33	**46.67**	–	6.67	20.00
Car	–	–	–	0.00	0.00	–	0.00	**43.33**	–	–	0.00	41.33	26.67	34.67	42.67	–	20.00	26.67
Cleveland	–	–	–	18.00	24.80	–	58.00	59.20	–	–	68.40	70.40	68.80	66.80	**72.00**	–	42.80	44.00
Dermatology	–	–	–	6.13	30.40	–	33.60	61.33	–	–	36.00	65.60	54.67	59.20	**66.13**	–	37.33	54.93
Ecoli	–	–	–	0.00	0.00	–	65.43	69.57	–	–	73.00	**77.57**	75.57	75.57	76.29	–	63.14	65.57
Glass	–	–	–	0.00	5.90	–	52.00	63.05	–	–	58.86	65.52	62.86	63.24	65.14	–	60.76	**66.67**
Hayes-roth	–	–	–	6.67	13.33	–	6.67	**33.33**	–	–	18.67	**33.33**	18.67	20.00	**33.33**	–	0.00	20.00
Iris	–	–	–	0.00	0.00	–	13.33	**60.00**	–	–	14.67	46.67	30.67	38.67	46.67	–	0.00	33.33
Lymphography	–	–	–	0.00	10.00	–	24.00	50.00	–	–	32.67	**54.00**	47.33	51.33	52.67	–	6.67	50.00
NewThyroid	–	–	–	0.00	0.00	–	20.00	33.33	–	–	17.33	33.33	16.00	18.67	**34.67**	–	0.00	20.00
Nursery	–	–	–	0.00	0.00	–	0.00	**70.00**	–	–	0.00	**70.00**	64.40	**70.00**	**70.00**	–	0.00	60.00
Page-blocks	–	–	–	0.00	35.20	–	46.80	51.20	–	–	48.00	**60.40**	55.60	59.60	59.20	–	51.20	56.00
Penbased	–	–	–	0.00	0.00	–	55.64	58.13	–	–	56.80	59.38	58.40	56.71	58.49	–	68.53	**68.80**
Satimage	–	–	–	0.00	0.57	–	51.05	58.29	–	–	62.67	65.14	64.19	64.76	**65.33**	–	54.86	59.05
Segment	–	–	–	0.00	0.57	–	40.00	53.14	–	–	47.62	59.62	56.76	56.95	**60.76**	–	54.86	58.10
Shuttle	–	–	–	0.00	20.00	–	40.40	**52.00**	–	–	39.20	48.00	46.80	48.00	48.40	–	15.60	50.00
Splice	–	–	–	0.00	0.00	–	0.00	**6.67**	–	–	1.33	4.00	0.00	0.00	1.33	–	0.00	0.00
Tae	–	–	–	0.00	0.00	–	26.67	**33.33**	–	–	13.33	13.33	13.33	13.33	14.67	–	0.00	0.00
Thyroid	–	–	–	0.00	0.00	–	26.67	**33.33**	–	–	20.00	26.67	24.00	22.67	24.00	–	20.00	26.67
Vowel	–	–	–	0.00	0.00	–	62.33	70.18	–	–	61.02	65.09	61.82	64.65	65.31	–	69.82	**70.91**
Wine	–	–	–	0.00	0.00	–	0.00	**33.33**	–	–	0.00	**33.33**	29.33	**33.33**	**33.33**	–	0.00	**33.33**
Zoo	–	–	–	0.00	41.90	–	43.24	**80.95**	–	–	43.81	74.48	67.24	74.29	75.62	–	18.10	73.33
Average	–	–	–	1.34	7.94	–	30.55	50.68	–	–	32.59	**50.70**	44.66	47.11	50.52	–	27.27	43.48
Instance Reduction																		
			WV			NMC with $k=1$										DT		
				CS			CS		IS		CS_IS						CS	
Data-set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	–	–	–	–	–	–	–	–	62.64	86.45	50.79	76.03	88.70	89.94	88.07	–	**95.49**	**95.49**
Balance	–	–	–	–	–	–	–	–	13.46	98.76	14.28	55.26	98.82	98.80	98.79	–	**99.40**	**99.40**
Car	–	–	–	–	–	–	–	–	0.00	99.42	0.00	54.96	99.42	99.42	99.39	–	**99.71**	**99.71**
Cleveland	–	–	–	–	–	–	–	–	92.46	93.57	92.56	92.83	93.89	93.55	93.71	–	**97.94**	**97.94**
Dermatology	–	–	–	–	–	–	–	–	56.22	95.49	56.45	82.58	95.81	95.80	95.67	–	**97.95**	**97.95**
Ecoli	–	–	–	–	–	–	–	–	90.22	92.18	89.87	91.43	93.37	93.40	93.34	–	**97.11**	**97.11**
Glass	–	–	–	–	–	–	–	–	79.94	90.82	81.05	84.89	91.72	91.56	91.65	–	**96.07**	**96.07**
Hayes-roth	–	–	–	–	–	–	–	–	67.28	92.92	70.07	72.97	92.73	92.73	92.05	–	**97.24**	**97.24**
Iris	–	–	–	–	–	–	–	–	54.30	95.00	56.90	60.33	95.00	95.00	94.83	–	**97.56**	**97.56**
Lymphography	–	–	–	–	–	–	–	–	65.71	92.52	64.05	65.55	93.21	93.17	93.10	–	**96.73**	**96.73**
NewThyroid	–	–	–	–	–	–	–	–	39.12	96.51	41.86	51.40	96.47	96.58	96.19	–	**98.29**	**98.29**
Nursery	–	–	–	–	–	–	–	–	0.00	99.23	0.00	64.96	99.23	99.23	99.23	–	**99.52**	**99.52**
Page-blocks	–	–	–	–	–	–	–	–	74.87	96.30	82.22	84.21	96.89	97.26	96.69	–	**98.87**	**98.87**
Penbased	–	–	–	–	–	–	–	–	77.93	87.03	82.10	85.45	90.93	90.16	88.70	–	**98.88**	**98.88**
Satimage	–	–	–	–	–	–	–	–	93.03	95.25	91.98	93.35	96.59	96.63	96.65	–	**98.66**	**98.66**
Segment	–	–	–	–	–	–	–	–	84.15	98.00	77.00	89.82	98.76	98.77	98.62	–	**99.62**	**99.62**
Shuttle	–	–	–	–	–	–	–	–	0.00	99.44	66.53	63.45	99.37	99.37	99.37	–	**99.71**	**99.71**
Splice	–	–	–	–	–	–	–	–	66.11	97.56	70.31	72.59	97.54	97.57	97.51	–	**98.84**	**98.84**
Tae	–	–	–	–	–	–	–	–	81.33	93.01	81.73	81.60	91.48	91.74	91.68	–	**97.58**	**97.58**

(*continued on next page*)

Table A.18 (*continued*)

Accuracy																		
			WV			NMC with $k = 1$										DT		
				CS			CS		IS		CS_IS						CS	
Data–set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Thyroid	–	–	–	–	–	–	–	–	67.12	98.86	70.86	70.42	98.72	98.73	98.71	–	**99.48**	**99.48**
Vowel	–	–	–	–	–	–	–	–	63.10	74.95	73.11	75.98	82.40	84.74	82.84	–	**98.63**	**98.63**
Wine	–	–	–	–	–	–	–	–	0.00	95.79	0.00	48.74	95.79	95.79	95.79	–	**97.94**	**97.94**
Zoo	–	–	–	–	–	–	–	–	45.93	82.90	45.48	74.11	82.66	82.66	82.66	–	**92.02**	**92.02**
Average	–	–	–	–	–	–	–	–	55.43	93.56	59.10	73.61	94.33	94.46	94.14	–	**97.97**	**97.97**
Training Times																		
Autos	–	–	–	0.54	0.35	–	2.74	2.71	2.71	4.02	3.12	2.68	3.22	3.16	3.13	–	0.41	**0.21**
Balance	–	–	–	0.24	0.09	–	1.82	1.67	33.76	4.29	28.96	16.84	4.54	4.57	4.05	–	0.47	**0.07**
Car	–	–	–	0.54	0.44	–	49.73	52.34	1415	77.69	899.5	488.1	103.4	105.7	121.5	–	0.20	**0.15**
Cleveland	–	–	–	0.52	0.35	–	3.77	3.58	7.20	7.52	7.28	7.32	6.94	6.83	6.39	–	0.39	**0.18**
Dermatology	–	–	–	0.65	0.65	–	8.62	11.70	12.04	7.48	9.85	6.90	7.31	7.79	10.04	–	**0.25**	0.26
Ecoli	–	–	–	1.59	1.36	–	25.73	28.14	26.36	27.82	15.42	16.18	18.82	17.68	17.56	–	**0.80**	0.81
Glass	–	–	–	1.13	0.89	–	7.68	7.53	6.61	7.24	6.00	5.91	6.14	6.45	6.73	–	0.51	**0.28**
Hayes-roth	–	–	–	0.14	**0.03**	–	0.22	0.15	0.42	0.50	0.39	0.36	0.39	0.44	0.46	–	0.12	0.04
Iris	–	–	–	0.13	0.04	–	0.20	0.18	0.52	0.40	0.42	0.39	0.34	0.37	0.35	–	0.12	**0.04**
Lymphography	–	–	–	0.24	0.11	–	0.48	0.45	0.71	0.64	0.73	0.67	0.63	0.62	0.64	–	0.20	**0.08**
NewThyroid	–	–	–	0.21	0.05	–	0.36	0.29	1.38	0.81	1.23	0.99	0.78	0.76	0.84	–	0.23	**0.05**
Nursery	–	–	–	1.72	1.62	–	29.85	51.29	878.2	29.01	577.1	214.3	35.85	37.12	40.62	–	**0.20**	0.21
Page-blocks	–	–	–	0.51	0.47	–	14.04	13.11	24.80	17.66	20.67	17.38	12.76	11.70	14.16	-	0.22	**0.15**
Penbased	–	–	–	**8.92**	8.99	–	1184	1137	1146	967.7	631.5	597.5	545.9	567.7	564.7	–	9.75	9.08
Satimage	–	–	–	2.08	1.98	–	67.70	67.48	75.27	73.58	41.94	36.96	45.99	48.25	48.82	–	0.79	**0.73**
Segment	–	–	–	10.04	8.89	–	949.2	1024	1871	874.3	1486	893.4	580.8	587.9	614.9	–	**2.42**	2.73
Shuttle	–	–	–	1.92	2.73	–	182.7	212.3	2426	159.9	1119	1058	199.1	200.7	214.0	–	**0.47**	0.49
Splice	–	–	–	0.17	**0.05**	–	0.68	0.61	2.63	2.14	2.53	2.50	2.13	2.22	2.30	–	0.18	0.05
Tae	–	–	–	0.13	**0.03**	–	0.24	0.16	0.70	0.85	0.78	0.84	0.90	1.02	0.98	–	0.14	0.04
Thyroid	–	–	–	0.17	0.10	–	2.25	2.40	23.94	8.04	23.50	22.50	9.99	9.77	9.45	–	0.12	**0.02**
Vowel	–	–	–	**8.65**	9.04	–	981.1	1108	1587	1367	801.4	772.0	730.3	681.0	715.3	–	13.33	12.64
Wine	–	–	–	0.12	**0.03**	–	0.27	0.19	1.09	0.49	0.88	0.51	0.37	0.42	0.41	–	0.14	0.04
Zoo	–	–	–	0.50	0.36	–	0.98	1.23	0.84	1.28	1.11	1.26	1.26	1.61	1.56	–	0.24	**0.23**
Average	–	–	–	1.78	1.68	–	152.8	162.0	414.9	158.3	246.9	181.0	100.8	100.2	104.3	–	1.38	**1.24**

Table A.19
Average accuracy results in test, classifier and instance reduction rates and training times for C4.5 classifier.

Accuracy																		
			WV			NMC with $k = 1$										DT		
				CS			CS		IS		CS_IS						CS	
Data–set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	76.24	74.96	76.24	76.24	76.24	75.61	**77.69**	77.12	76.68	75.86	73.33	73.78	75.64	73.93	73.54	67.95	77.15	76.65
Balance	80.63	**81.59**	80.63	80.63	80.63	50.03	50.03	50.03	80.63	80.63	80.63	80.63	80.63	80.63	80.63	80.63	80.63	80.63
Car	94.68	94.50	94.68	94.68	94.68	93.00	92.94	92.94	94.76	94.73	94.55	94.53	**94.79**	94.71	94.65	91.95	93.51	93.52
Cleveland	52.55	53.55	52.55	52.22	51.86	53.93	53.99	**54.26**	53.43	52.48	53.28	52.87	53.26	52.93	52.93	51.52	49.38	49.51
Dermatology	95.24	**98.32**	95.24	95.24	95.79	95.52	95.75	95.30	96.03	96.31	95.81	95.26	95.42	95.19	95.48	96.63	95.53	95.08
Ecoli	81.06	81.94	81.06	81.01	80.89	77.78	77.75	77.40	82.48	82.10	82.86	82.86	82.87	**82.88**	82.63	80.56	80.52	79.98
Glass	72.03	71.63	**72.03**	**72.03**	**72.03**	68.19	68.14	68.14	70.00	70.64	70.62	70.73	70.89	71.94	71.10	66.09	69.63	69.73
Hayes-roth	84.12	84.12	**84.12**	**84.12**	83.35	73.52	73.52	73.52	**84.12**	**84.12**	**84.12**	**84.12**	**84.12**	**84.12**	**84.12**	**84.12**	**84.12**	**84.12**
Iris	94.67	**95.33**	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67
Lymphography	74.50	**76.44**	74.50	74.50	75.19	68.80	68.80	68.11	75.05	75.19	75.46	75.19	75.19	75.19	75.19	61.46	64.98	64.98
NewThyroid	91.16	93.02	91.16	91.16	91.16	71.16	71.16	71.16	92.56	**93.02**	91.81	90.70	90.70	90.70	90.70	90.70	90.70	90.70
Nursery	89.66	**89.81**	89.66	89.66	89.66	78.46	78.46	78.46	89.66	89.66	89.66	89.66	89.72	89.69	89.69	82.55	82.55	82.55
Page-blocks	95.64	95.46	95.64	95.64	95.64	95.64	95.17	95.21	**95.82**	95.60	95.78	95.78	**95.82**	**95.82**	**95.82**	92.40	95.12	94.91
Penbased	91.10	91.11	91.10	91.10	91.10	**93.37**	91.59	91.57	92.55	92.39	91.73	91.95	91.63	91.35	91.81	83.47	89.47	89.31
Satimage	82.15	**82.92**	82.15	82.15	82.15	82.31	82.00	82.00	82.18	81.74	81.32	81.45	81.19	81.39	81.04	81.51	81.21	81.21
Segment	96.28	96.71	96.28	96.28	96.26	**97.45**	97.03	96.93	97.39	97.07	96.95	96.78	96.69	96.74	96.75	93.94	95.92	96.10
Shuttle	99.59	99.68	99.59	99.59	99.60	**99.72**	99.70	**99.72**	99.63	99.59	99.61	99.63	99.62	99.63	99.63	99.63	99.63	99.67
Splice	89.69	90.61	89.69	89.69	89.69	**90.62**	**90.62**	90.30	**90.62**	90.56	90.49	90.56	90.49	90.56	**90.62**	89.71	**90.62**	**90.62**
Tae	54.77	54.77	**54.77**	**54.77**	**54.77**	48.95	49.61	49.61	53.38	**54.77**	53.94	53.81	**54.77**	**54.77**	**54.77**	54.13	54.13	53.46
Thyroid	98.89	96.53	98.89	98.89	98.89	98.75	98.75	98.75	98.89	98.89	98.92	98.92	**98.97**	**98.97**	**98.97**	98.89	98.89	98.89
Vowel	83.43	83.64	83.43	83.43	83.43	**90.91**	89.31	88.38	88.00	87.66	86.53	86.38	87.13	86.53	86.69	72.12	83.01	83.17
Wine	92.71	**94.98**	92.71	92.71	92.71	93.25	93.25	93.30	93.25	93.25	93.25	93.30	92.94	93.30	93.30	93.28	93.28	93.87
Zoo	92.17	92.17	92.17	92.17	92.17	92.09	91.42	91.04	**93.22**	**93.22**	91.31	90.80	90.44	90.79	90.60	93.04	89.94	90.16
Average	85.35	85.82	85.35	85.33	85.33	81.90	81.80	81.65	**85.87**	85.83	85.51	85.41	85.55	85.50	85.45	82.65	84.11	84.06
Classifier Reduction																		
Autos	–	–	–	0.00	1.07	–	29.60	42.67	–	–	45.07	**52.00**	46.93	49.60	48.53	–	40.27	45.33
Balance	–	–	–	0.00	33.33	–	0.00	**66.67**	–	–	49.33	**66.67**	62.67	**66.67**	**66.67**	–	0.00	**66.67**
Car	–	–	–	0.00	0.00	–	6.67	6.67	–	–	19.33	**25.33**	21.33	22.00	23.33	–	22.67	23.33

(*continued on next page*)

Table A.19 (*continued*)

Data–set	Comb	DCS	WV			NMC with $k=1$										DT		
				CS			CS		IS		CS_IS						CS	
				f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Accuracy																		
Cleveland	–	–	–	2.00	6.00	–	23.60	27.60	–	–	37.60	**40.80**	38.40	38.00	**40.80**	–	18.80	26.00
Dermatology	–	–	–	0.00	19.47	–	38.13	**64.00**	–	–	44.00	63.20	57.07	59.47	61.87	–	26.93	57.33
Ecoli	–	–	–	1.29	4.29	–	52.29	65.00	–	–	67.86	**75.71**	74.71	73.57	75.14	–	51.86	68.71
Glass	–	–	–	0.00	2.10	–	43.81	59.81	–	–	57.90	66.10	63.43	64.00	63.62	–	56.57	**67.43**
Hayes-roth	–	–	–	0.00	6.67	–	0.00	**33.33**	–	–	25.33	**33.33**	25.33	**33.33**	**33.33**	–	0.00	**33.33**
Iris	–	–	–	0.00	0.00	–	0.00	**33.33**	–	–	8.00	**33.33**	32.00	**33.33**	**33.33**	–	0.00	**33.33**
Lymphography	–	–	–	0.00	3.33	–	0.00	53.33	–	–	44.00	56.67	52.67	56.67	56.67	–	27.33	**60.00**
NewThyroid	–	–	–	0.00	0.00	–	6.67	26.67	–	–	9.33	**33.33**	**33.33**	**33.33**	**33.33**	–	6.67	**33.33**
Nursery	–	–	–	0.00	17.20	–	0.00	70.00	–	–	58.40	**80.00**	75.60	**80.00**	**80.00**	–	0.00	70.00
Page-blocks	–	–	–	0.00	9.60	–	34.00	50.00	–	–	38.80	52.40	49.60	51.20	51.60	–	45.60	**58.00**
Penbased	–	–	–	0.00	0.00	–	57.87	57.42	–	–	52.27	53.69	52.62	52.89	52.27	–	66.04	**66.67**
Satimage	–	–	–	0.00	0.00	–	46.86	53.33	–	–	52.95	**56.57**	54.48	54.10	54.67	–	48.57	53.33
Segment	–	–	–	0.00	1.14	–	35.43	42.29	–	–	44.57	**49.90**	47.43	46.29	47.05	–	48.38	48.00
Shuttle	–	–	–	0.00	12.00	–	18.80	58.00	–	–	16.80	63.60	56.00	59.20	**64.00**	–	6.80	56.00
Splice	–	–	–	0.00	0.00	–	0.00	**13.33**	–	–	2.67	12.00	6.67	6.67	10.67	–	6.67	6.67
Tae	–	–	–	0.00	0.00	–	6.67	**13.33**	–	–	0.00	1.33	0.00	0.00	0.00	–	0.00	6.67
Thyroid	–	–	–	0.00	13.33	–	0.00	**20.00**	–	–	9.33	13.33	13.33	13.33	13.33	–	13.33	**20.00**
Vowel	–	–	–	0.00	0.00	–	54.25	57.09	–	–	47.42	47.56	45.16	48.00	46.47	–	**62.91**	**62.91**
Wine	–	–	–	0.00	0.00	–	0.00	20.00	–	–	0.00	20.00	12.00	20.00	20.00	–	0.00	**26.67**
Zoo	–	–	–	0.00	0.00	–	53.52	**74.29**	–	–	49.71	71.81	63.62	70.10	72.57	–	43.24	69.52
Average	–	–	–	0.14	5.63	–	22.09	43.83	–	–	33.94	**46.46**	42.80	44.86	45.62	–	25.77	46.05
Instance Reduction																		
Autos	–	–	–	–	–	–	–	–	73.84	83.59	74.75	78.99	85.82	86.51	85.90	–	**95.49**	**95.49**
Balance	–	–	–	–	–	–	–	–	71.81	99.20	72.56	74.84	99.20	99.20	99.20	–	**99.40**	**99.40**
Car	–	–	–	–	–	–	–	–	80.64	99.17	79.86	79.87	99.17	99.09	99.04	–	**99.71**	**99.71**
Cleveland	–	–	–	–	–	–	–	–	91.23	92.78	90.55	91.52	93.50	93.54	93.35	–	**97.94**	**97.94**
Dermatology	–	–	–	–	–	–	–	–	49.98	95.73	63.11	75.58	95.28	95.50	95.12	–	**97.95**	**97.95**
Ecoli	–	–	–	–	–	–	–	–	88.39	93.11	87.57	91.29	94.54	94.39	94.40	–	**97.11**	**97.11**
Glass	–	–	–	–	–	–	–	–	83.66	89.86	82.24	85.18	91.14	91.33	90.28	–	**96.07**	**96.07**
Hayes-roth	–	–	–	–	–	–	–	–	54.91	94.32	60.69	62.77	94.32	94.32	94.32	–	**97.24**	**97.24**
Iris	–	–	–	–	–	–	–	–	51.07	95.00	52.60	48.00	95.00	95.00	95.00	–	**97.56**	**97.56**
Lymphography	–	–	–	–	–	–	–	–	61.83	93.92	71.02	75.20	94.60	94.66	94.60	–	**96.73**	**96.73**
NewThyroid	–	–	–	–	–	–	–	–	30.63	95.93	30.49	49.00	96.51	96.51	96.51	–	**98.29**	**98.29**
Nursery	–	–	–	–	–	–	–	–	76.03	99.42	76.70	83.41	99.42	99.42	99.42	–	**99.52**	**99.52**
Page-blocks	–	–	–	–	–	–	–	–	57.41	96.53	60.47	65.11	97.35	97.29	97.26	–	**98.87**	**98.87**
Penbased	–	–	–	–	–	–	–	–	83.13	87.40	83.74	86.06	89.33	89.64	87.93	–	**98.88**	**98.88**
Satimage	–	–	–	–	–	–	–	–	91.29	94.67	90.04	90.51	94.74	94.62	94.30	–	**98.66**	**98.66**
Segment	–	–	–	–	–	–	–	–	86.21	97.03	86.46	89.99	96.66	96.90	96.16	–	**99.62**	**99.62**
Shuttle	–	–	–	–	–	–	–	–	20.01	99.47	21.15	66.62	99.47	99.47	99.43	–	**99.71**	**99.71**
Splice	–	–	–	–	–	–	–	–	52.47	97.65	59.55	61.79	97.65	97.65	97.51	–	**98.84**	**98.84**
Tae	–	–	–	–	–	–	–	–	75.58	94.77	79.70	74.77	94.87	94.90	94.83	–	**97.58**	**97.58**
Thyroid	–	–	–	–	–	–	–	–	43.21	98.89	45.60	45.90	98.96	98.96	98.96	–	**99.48**	**99.48**
Vowel	–	–	–	–	–	–	–	–	61.84	64.67	65.38	63.71	67.46	69.22	64.92	–	**98.63**	**98.63**
Wine	–	–	–	–	–	–	–	–	0.00	95.76	0.00	29.68	95.76	95.73	95.73	–	**97.94**	**97.94**
Zoo	–	–	–	–	–	–	–	–	53.38	82.44	52.80	71.36	82.56	82.56	82.26	–	**92.02**	**92.02**
Average	–	–	–	–	–	–	–	–	62.55	93.10	64.65	71.35	93.62	93.76	93.32	–	**97.97**	**97.97**
Training Times																		
Autos	–	–	–	0.21	0.17	–	2.27	2.53	2.54	3.18	2.60	2.69	3.00	3.19	3.60	–	0.21	**0.15**
Balance	–	–	–	0.06	0.03	–	0.79	1.05	11.05	2.84	11.05	9.99	2.96	2.99	3.03	–	0.11	**0.01**
Car	–	–	–	0.25	0.21	–	34.88	36.76	208.4	143.8	204.8	204.8	141.5	143.5	147.8	–	0.20	**0.15**
Cleveland	–	–	–	0.21	0.17	–	3.90	4.07	6.34	6.50	6.33	6.25	6.92	6.76	7.06	–	0.22	**0.11**
Dermatology	–	–	–	0.31	0.37	–	4.64	7.71	8.17	4.05	8.53	6.61	6.14	6.50	7.25	–	**0.19**	0.22
Ecoli	–	–	–	0.67	**0.59**	–	22.00	24.17	13.80	15.12	12.98	12.77	14.09	15.25	15.67	–	0.61	0.67
Glass	–	–	–	0.43	0.31	–	5.93	6.40	4.62	5.35	5.42	5.26	6.15	6.56	6.49	–	0.32	**0.21**
Hayes-roth	–	–	–	0.06	0.01	–	0.08	0.05	0.19	0.19	0.20	0.19	0.20	0.21	0.21	–	0.03	**0.01**
Iris	–	–	–	0.05	0.02	–	0.09	0.08	0.26	0.20	0.27	0.27	0.21	0.22	0.22	–	0.03	**0.01**
Lymphography	–	–	–	0.07	0.03	–	0.20	0.20	0.33	0.37	0.40	0.36	0.29	0.29	0.30	–	0.04	**0.02**
NewThyroid	–	–	–	0.05	0.02	–	0.16	0.13	0.97	0.53	0.93	0.78	0.43	0.43	0.44	–	0.04	**0.01**
Nursery	–	–	–	0.50	0.71	–	20.94	31.92	123.7	24.67	110.9	85.77	26.02	27.39	29.22	–	**0.18**	0.21
Page-blocks	–	–	–	0.19	0.15	–	7.33	8.65	22.32	10.11	21.15	17.97	8.24	8.38	8.72	–	0.14	**0.11**
Penbased	–	–	–	5.91	**5.90**	–	1239	1437	655.5	596.9	633.4	602.5	554.3	542.4	567.5	–	14.07	17.45
Satimage	–	–	–	**0.91**	0.93	–	61.76	60.36	56.86	55.78	57.25	54.05	54.92	55.74	58.15	–	1.03	1.10
Segment	–	–	–	**3.94**	4.25	–	1163	1352	1361	812.6	1051	937.4	656.4	643.0	726.7	–	5.26	5.80
Shuttle	–	–	–	0.76	0.64	–	93.02	134.3	1701	81.76	1386	895.1	111.2	112.9	137.5	–	**0.31**	0.46
Splice	–	–	–	0.06	0.03	–	0.39	0.33	2.46	1.05	2.56	2.48	1.21	1.19	1.23	–	0.04	**0.01**
Tae	–	–	–	0.05	0.01	–	0.11	0.07	0.35	0.46	0.41	0.42	0.49	0.46	0.54	–	0.03	**0.01**
Thyroid	–	–	–	0.09	0.05	–	1.60	1.71	32.91	4.92	29.10	28.40	5.55	5.73	6.07	–	0.06	**0.02**
Vowel	–	–	–	7.59	**7.48**	–	1904	1956	1114	1145	1076	1050	1053	1034	1061	–	30.34	34.36
Wine	–	–	–	0.05	0.01	–	0.12	0.08	0.76	0.39	0.69	0.55	0.40	0.42	0.41	–	0.03	**0.01**
Zoo	–	–	–	0.18	0.11	–	0.61	1.08	0.47	0.70	0.98	1.21	1.31	1.50	1.73	–	**0.09**	0.12
Average	–	–	–	0.98	**0.97**	–	198.6	220.3	231.7	126.8	200.9	170.7	115.4	113.9	121.3	–	2.33	2.66

Table A.20
Average accuracy results in test, classifier and instance reduction rates and training times for Ripper classifier.

Accuracy																		
			WV			NMC with $k = 1$										DT		
				CS			CS		IS		CS_IS						CS	
Data-set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	**85.09**	84.42	**85.09**	**85.09**	84.48	84.42	84.51	83.55	82.10	81.27	83.18	82.33	82.23	82.83	82.73	80.69	82.77	83.23
Balance	78.54	78.22	78.54	78.54	78.54	**80.95**	80.63	80.63	80.82	80.85	80.82	80.92	80.88	80.95	80.76	78.22	78.22	78.22
Car	92.59	**93.52**	92.59	92.59	92.59	90.97	90.85	90.85	92.71	92.73	92.71	92.93	92.81	92.78	92.83	92.13	92.13	92.01
Cleveland	52.18	**54.54**	52.18	51.84	51.84	53.49	52.62	51.76	52.76	51.01	50.54	51.41	51.63	51.55	51.49	50.46	49.59	51.71
Dermatology	93.32	**94.43**	93.32	93.32	93.11	93.33	92.88	93.06	93.68	94.01	93.67	94.01	93.71	93.44	93.45	93.33	93.38	93.83
Ecoli	78.47	78.74	78.47	78.47	78.47	74.61	75.30	75.35	78.96	78.95	78.59	78.47	78.94	79.49	79.17	**79.66**	78.77	78.58
Glass	68.56	68.12	68.56	68.56	68.56	**70.80**	66.83	66.17	69.60	70.38	69.42	68.75	68.62	68.76	68.16	67.49	67.76	67.48
Hayes-roth	83.41	83.41	83.41	83.41	83.41	**84.89**	**84.89**	**84.89**	83.38	82.92	83.82	83.51	83.09	82.93	83.07	83.35	83.35	83.35
Iris	93.33	**95.33**	93.33	93.33	93.33	94.67	94.67	94.27	94.40	94.00	94.13	94.00	94.67	94.67	94.67	94.00	94.00	93.60
Lymphography	**75.68**	**75.68**	**75.68**	**75.68**	**75.68**	70.13	69.46	69.46	74.17	73.62	73.06	73.07	72.66	73.21	73.21	75.05	75.05	75.05
NewThyroid	92.09	**93.49**	92.09	92.09	92.09	92.09	92.09	92.09	92.09	92.09	92.09	92.09	92.09	92.09	92.09	92.09	92.09	92.09
Nursery	90.66	**90.81**	90.66	90.66	90.51	90.43	90.37	90.27	90.65	90.80	90.51	90.48	90.58	90.58	90.55	88.52	88.54	88.44
Page-blocks	95.45	95.11	95.45	95.45	95.45	94.92	95.10	**95.63**	95.31	95.24	95.42	95.35	95.38	95.31	95.45	94.54	94.92	94.99
Penbased	91.38	91.11	91.38	91.38	91.38	**94.19**	92.57	92.88	92.42	92.24	91.96	91.57	92.19	91.97	92.17	84.20	88.12	88.30
Satimage	82.61	82.14	82.61	82.61	82.61	**82.77**	81.20	80.96	81.42	81.21	81.79	81.30	81.70	81.45	81.48	81.19	80.85	80.76
Segment	96.58	96.88	96.54	96.54	96.54	**96.88**	96.76	96.80	96.73	96.78	96.72	96.68	96.70	96.53	96.61	93.60	96.02	96.09
Shuttle	99.40	**99.68**	99.40	99.40	99.45	99.59	99.61	99.55	99.51	99.50	99.51	99.51	99.53	99.51	99.47	99.17	99.51	99.51
Splice	88.11	**90.30**	88.11	88.11	88.11	87.46	87.46	87.46	86.90	87.72	87.09	87.03	87.40	87.40	87.65	89.36	89.36	89.36
Tae	55.50	55.57	55.50	55.50	55.50	57.99	56.66	56.66	**58.14**	57.37	57.22	57.35	57.13	57.39	56.86	56.06	56.06	56.75
Thyroid	97.78	96.25	97.78	97.78	97.78	97.64	97.64	97.78	98.06	97.84	98.12	**98.20**	98.09	**98.20**	97.97	95.85	97.92	97.92
Vowel	80.20	79.39	79.60	79.60	79.60	**91.01**	86.77	86.85	87.43	87.39	86.91	86.57	86.77	85.92	86.40	73.54	81.35	80.44
Wine	96.68	**97.22**	96.68	96.68	96.68	96.09	96.09	96.09	96.09	97.12	96.09	96.09	96.23	96.23	96.44	96.68	96.68	96.68
Zoo	94.05	94.05	94.05	94.05	94.05	96.10	95.05	**96.38**	96.10	96.10	95.22	94.58	94.88	94.15	94.05	93.88	95.05	93.22
Average	85.29	85.58	85.26	85.25	85.21	**85.89**	85.22	85.19	85.80	85.70	85.59	85.49	85.56	85.54	85.51	84.05	84.85	84.85
Classifier Reduction																		
Autos	–	–	–	0.00	1.87	–	36.80	45.87	–	–	45.33	50.40	51.20	50.67	**52.53**	–	39.47	46.93
Balance	–	–	–	0.00	0.00	–	**6.67**	**6.67**	–	–	0.00	0.00	0.00	0.00	0.00	–	0.00	0.00
Car	–	–	–	0.00	0.00	–	3.33	3.33	–	–	15.33	**22.00**	18.00	18.00	22.00	–	0.00	3.33
Cleveland	–	–	–	2.00	2.00	–	26.00	**34.80**	–	–	28.40	30.00	30.00	30.40	30.40	–	14.00	20.00
Dermatology	–	–	–	0.00	4.53	–	34.13	47.73	–	–	37.60	**57.07**	43.20	46.13	54.93	–	41.60	48.80
Ecoli	–	–	–	0.00	0.00	–	59.86	62.00	–	–	70.00	**74.43**	72.14	72.29	73.86	–	59.86	63.71
Glass	–	–	–	0.00	0.00	–	53.71	**65.71**	–	–	61.33	65.14	64.19	65.33	**65.71**	–	54.29	61.90
Hayes-roth	–	–	–	0.00	0.00	–	0.00	**33.33**	–	–	28.00	**33.33**	28.00	**33.33**	**33.33**	–	0.00	**33.33**
Iris	–	–	–	0.00	0.00	–	0.00	**33.33**	–	–	4.00	**33.33**	32.00	**33.33**	**33.33**	–	0.00	**33.33**
Lymphography	–	–	–	0.00	0.00	–	10.67	33.33	–	–	28.00	54.67	39.33	55.33	**56.67**	–	0.00	50.00
NewThyroid	–	–	–	0.00	0.00	–	0.00	**26.67**	–	–	0.00	**26.67**	**26.67**	**26.67**	**26.67**	–	0.00	**26.67**
Nursery	–	–	–	0.00	6.00	–	23.20	48.00	–	–	41.60	**62.00**	57.60	61.60	61.60	–	33.20	48.00
Page-blocks	–	–	–	0.00	0.00	–	14.00	50.00	–	–	22.80	52.40	46.40	49.60	52.80	–	35.20	**56.00**
Penbased	–	–	–	0.00	0.00	–	54.22	55.47	–	–	45.96	47.64	46.40	46.04	49.42	–	**64.27**	**64.27**
Satimage	–	–	–	0.00	0.00	–	44.76	52.57	–	–	56.19	58.10	57.33	58.67	**59.43**	–	50.48	53.90
Segment	–	–	–	0.00	0.00	–	39.81	47.62	–	–	42.10	48.00	45.14	44.95	**48.57**	–	47.81	48.19
Shuttle	–	–	–	0.00	17.20	–	20.40	52.00	–	–	36.40	60.00	52.00	60.80	**61.60**	–	18.00	56.00
Splice	–	–	–	**0.00**	**0.00**	–	**0.00**	**0.00**	–	–	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	–	**0.00**	**0.00**
Tae	–	–	–	0.00	0.00	–	**6.67**	**6.67**	–	–	0.00	0.00	0.00	0.00	0.00	–	0.00	**6.67**
Thyroid	–	–	–	6.67	13.33	–	0.00	**26.67**	–	–	8.00	**26.67**	**26.67**	**26.67**	**26.67**	–	20.00	**26.67**
Vowel	–	–	–	0.00	0.00	–	56.65	58.40	–	–	45.67	46.47	43.71	45.96	45.45	–	65.02	**65.60**
Wine	–	–	–	0.00	0.00	–	0.00	**13.33**	–	–	0.00	**13.33**	6.67	6.67	**13.33**	–	0.00	**13.33**
Zoo	–	–	–	0.00	2.29	–	37.71	70.48	–	–	40.19	**73.14**	60.95	68.76	72.76	–	34.29	69.52
Average	–	–	–	0.38	2.05	–	22.98	38.00	–	–	28.56	40.64	36.85	39.18	**40.92**	–	25.11	38.96
Instance Reduction																		
			WV			NMC with $k = 1$										DT		
				CS			CS		IS		CS_IS						CS	
Data–set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Autos	–	–	–	–	–	–	–	–	69.36	82.76	72.70	76.16	89.23	87.91	88.59	–	**95.49**	**95.49**
Balance	–	–	–	–	–	–	–	–	90.94	97.60	91.79	91.88	97.50	97.52	97.52	–	**99.40**	**99.40**
Car	–	–	–	–	–	–	–	–	85.86	99.22	87.27	84.91	99.18	99.22	99.16	–	**99.71**	**99.71**
Cleveland	–	–	–	–	–	–	–	–	90.10	91.78	90.34	89.80	92.32	92.54	92.34	–	**97.94**	**97.94**
Dermatology	–	–	–	–	–	–	–	–	71.09	94.94	65.71	83.44	95.71	95.71	95.45	–	**97.95**	**97.95**
Ecoli	–	–	–	–	–	–	–	–	90.81	92.19	91.16	91.31	93.98	94.12	93.68	–	**97.11**	**97.11**
Glass	–	–	–	–	–	–	–	–	82.01	88.50	84.74	86.32	90.31	90.42	89.48	–	**96.07**	**96.07**
Hayes-roth	–	–	–	–	–	–	–	–	59.95	94.32	61.58	65.13	94.32	94.32	94.32	–	**97.24**	**97.24**
Iris	–	–	–	–	–	–	–	–	20.30	95.00	20.23	50.00	95.00	95.00	95.00	–	**97.56**	**97.56**
Lymphography	–	–	–	–	–	–	–	–	49.65	93.93	58.08	78.83	93.93	93.93	93.93	–	**96.73**	**96.73**
NewThyroid	–	–	–	–	–	–	–	–	0.00	96.51	0.00	39.16	96.51	96.51	96.51	–	**98.29**	**98.29**
Nursery	–	–	–	–	–	–	–	–	66.34	98.83	64.03	71.58	99.01	98.92	98.92	–	**99.52**	**99.52**
Page-blocks	–	–	–	–	–	–	–	–	28.12	97.06	37.16	71.67	97.48	96.49	97.16	–	**98.87**	**98.87**
Penbased	–	–	–	–	–	–	–	–	82.68	86.50	81.82	81.28	83.61	83.91	85.45	–	**98.88**	**98.88**
Satimage	–	–	–	–	–	–	–	–	90.17	93.84	88.90	90.75	94.08	93.56	93.88	–	**98.66**	**98.66**
Segment	–	–	–	–	–	–	–	–	86.60	97.13	82.38	86.90	97.45	97.47	97.79	–	**99.62**	**99.62**
Shuttle	–	–	–	–	–	–	–	–	49.93	99.41	57.80	72.69	99.43	99.43	99.40	–	**99.71**	**99.71**
Splice	–	–	–	–	–	–	–	–	51.79	97.49	64.54	63.78	97.52	97.52	97.49	–	**98.84**	**98.84**
Tae	–	–	–	–	–	–	–	–	84.59	92.00	84.06	83.96	92.53	92.69	92.00	–	**97.58**	**97.58**

(*continued on next page*)

Table A.20 (*continued*)

Accuracy																		
			WV			NMC with $k = 1$										DT		
				CS			CS		IS		CS_IS						CS	
Data-set	Comb	DCS		f_A	f_{AC}		f_A	f_{AC}	f_A	f_{AI}	f_A	f_{AC}	f_{AI}	f_{AIC}	f_{ACI}		f_A	f_{AC}
Thyroid	–	–	–	–	–	–	–	–	20.01	98.80	20.52	41.89	98.92	98.91	98.92	–	**99.48**	**99.48**
Vowel	–	–	–	–	–	–	–	–	65.26	66.31	65.71	66.18	66.15	66.94	66.61	–	**98.63**	**98.63**
Wine	–	–	–	–	–	–	–	–	10.33	95.79	9.47	29.02	95.79	95.79	95.50	–	**97.94**	**97.94**
Zoo	–	–	–	–	–	–	–	–	22.56	82.29	47.19	68.66	82.45	82.41	81.37	–	**92.02**	**92.02**
Average	–	–	–	–	–	–	–	–	59.50	92.70	62.05	72.41	93.15	93.10	93.06	–	**97.97**	**97.97**
Training Times																		
Autos	–	–	–	0.21	0.15	–	2.26	2.44	2.79	3.81	2.26	2.34	2.94	2.95	3.08	–	0.20	**0.13**
Balance	–	–	–	0.08	0.04	–	1.28	1.23	12.06	9.04	12.96	12.65	10.74	10.88	10.77	–	0.13	**0.02**
Car	–	–	–	0.29	0.22	–	42.22	41.42	159.6	141.7	161.8	164.7	150.4	147.5	151.0	–	0.19	**0.17**
Cleveland	–	–	–	0.21	0.13	–	3.94	3.77	7.95	7.38	7.63	6.81	8.21	8.35	8.30	–	0.23	**0.12**
Dermatology	–	–	–	0.33	0.39	–	11.47	12.91	7.46	6.97	10.42	9.09	8.73	9.67	11.56	–	0.33	**0.32**
Ecoli	–	–	–	0.76	**0.69**	–	30.54	34.23	16.86	17.14	16.27	17.58	18.54	18.23	18.94	–	0.81	0.89
Glass	–	–	–	0.39	0.27	–	5.54	6.08	6.28	7.94	6.13	6.42	7.15	6.95	7.31	–	0.32	**0.24**
Hayes-roth	–	–	–	0.05	0.02	–	0.09	0.06	0.20	0.18	0.22	0.21	0.20	0.21	0.22	–	0.03	**0.01**
Iris	–	–	–	0.05	0.01	–	0.09	0.08	0.38	0.19	0.38	0.27	0.21	0.21	0.22	–	0.03	**0.01**
Lymphography	–	–	–	0.07	0.03	–	0.30	0.28	0.34	0.22	0.53	0.42	0.31	0.36	0.37	–	0.04	**0.02**
NewThyroid	–	–	–	0.06	0.02	–	0.19	0.17	1.41	0.49	1.27	0.90	0.46	0.45	0.46	–	0.04	**0.01**
Nursery	–	–	–	0.66	0.73	–	43.94	59.86	211.7	63.06	204.0	143.5	51.83	53.43	54.48	–	**0.35**	0.37
Page-blocks	–	–	–	0.25	0.17	–	7.76	8.96	28.37	9.11	28.05	19.68	10.35	11.71	11.66	–	0.16	**0.11**
Penbased	–	–	–	6.00	**5.88**	–	1540	1281	651.8	593.2	658.1	649.7	633.3	624.8	617.5	–	18.25	21.84
Satimage	–	–	–	1.11	**1.01**	–	63.20	69.85	70.21	65.58	56.54	57.05	61.02	61.41	64.22	–	1.04	1.16
Segment	–	–	–	4.13	**3.94**	–	1028	1096	1248	764.9	1180	1031	638.1	633.9	686.9	–	4.81	4.82
Shuttle	–	–	–	0.85	1.04	–	110.9	148.6	1226	79.92	1019	689.6	121.5	122.2	137.1	–	**0.34**	0.53
Splice	–	–	–	0.06	0.03	–	0.39	0.34	2.54	1.33	2.41	2.41	1.50	1.53	1.54	–	0.04	**0.01**
Tae	–	–	–	0.06	0.01	–	0.12	0.08	0.56	0.71	0.63	0.57	0.71	0.76	0.79	–	0.03	**0.01**
Thyroid	–	–	–	0.09	0.05	–	1.46	1.62	40.66	5.68	38.25	30.98	5.44	5.60	5.70	–	0.06	**0.02**
Vowel	–	–	–	**8.88**	9.47	–	1753	1911	1052	1032	1036	1031	1048	1026	1052	–	25.22	22.74
Wine	–	–	–	0.05	0.01	–	0.13	0.09	0.75	0.34	0.66	0.56	0.40	0.41	0.38	–	0.03	**0.01**
Zoo	–	–	–	0.19	0.15	–	0.62	1.11	0.54	0.72	0.99	1.35	1.26	1.53	1.74	–	**0.09**	0.10
Average	–	–	–	1.08	**1.06**	–	202.0	203.5	206.4	122.2	193.2	168.7	120.9	119.5	123.7	–	2.29	2.33

References

[1] J.A. Sanz, M. Galar, A. Jurío, A. Brugos, M. Pagola, H. Bustince, Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system, Appl. Soft Comput. 20 (2014) 103–111.

[2] M. Galar, J. Derrac, D. Peralta, I. Triguero, D. Paternain, C. Lopez-Molina, S. García, J.M. Ben'ıtez, M. Pagola, E. Barrenechea, H. Bustince, F. Herrera, A survey of fingerprint classification part II: experimental analysis and ensemble proposal, Knowl.-based Syst. 81 (2015) 98–116.

[3] V. Bolón-Canedo, N.S.-M. no, A. Alonso-Betanzos, An ensemble of filters and classifiers for microarray data classification, Pattern Recogn. 45 (1) (2012) 531–539.

[4] I. Katakis, G. Tsoumakas, I. Vlahavas, Tracking recurring contexts using ensemble classifiers: an application to email filtering, Knowl. Inf. Syst. 22 (2010) 371–391.

[5] V. Vapnik, Statistical Learning Theory, New York: Wiley, 1998.

[6] A.C. Lorena, A.C. Carvalho, J.M. Gama, A review on the combination of binary classifiers in multiclass problems, Artif. Intell. Rev. 30 (1–4) (2008) 19–37.

[7] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience (2004).

[8] S. Knerr, L. Personnaz, G. Dreyfus, Single-layer learning revisited: a stepwise procedure for building and training a neural network, in: F.F. Soulié, J. Hérault (Eds.), Neurocomputing: Algorithms, Architectures and Applications, Vol. F68 of NATO ASI Series, Springer-Verlag, 1990, pp. 41–50.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, SIGKDD Explor. Newsl. 11 (2009) 10–18.

[10] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[11] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J. Fernández, F. Herrera, KEEL: A software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (3) (2009) 307–318.

[12] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artif. Intell. Res. 2 (1995) 263–286.

[13] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2000) 113–141.

[14] E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting, Pattern Recogn. 43 (1) (2010) 128–142.

[15] M. Elkano, M. Galar, J. Sanz, A. Fernandez, E. Barrenechea, F. Herrera, H. Bustince, Enhancing multi-class classification in farc-hd fuzzy classifier: on the synergy between n-dimensional overlap functions and decomposition strategies, IEEE Trans. Fuzzy Syst. 23 (5) (2015) 1562–1580.

[16] J. Fürnkranz, Round robin classification, J. Mach. Learn. Res. 2 (2002) 721–747.

[17] J. Fürnkranz, Round robin ensembles, Intell. Data Anal. 7 (5) (2003) 385–403.

[18] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, 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) (2011) 1761–1776.

[19] J.A. Sáez, M. Galar, J. Luengo, F. Herrera, Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition, Knowl. Inf. Syst. 38 (1) (2014) 179–206.

[20] B. Fei, J. Liu, Binary tree of SVM: a new fast multiclass training and classification algorithm, IEEE Trans. Neural Networks 17 (3) (2006) 696–704.

[21] B. Liu, Z. Hao, E.C.C. Tsang, Nesting one-against-one algorithm based on SVMs for pattern classification, IEEE Trans. Neural Networks 19 (12) (2008) 2044–2052.

[22] J.C. Platt, N. Cristianini, J. Shawe-taylor, Large margin DAGs for multiclass classification, in: Advances in Neural Information Processing Systems, MIT Press, 2000, pp. 547–553.

[23] T.F. Wu, C.J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (2004) 975–1005.

[24] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Dynamic classifier selection for one-vs-one strategy: avoiding non-competent classifiers, Pattern Recogn. 46 (12) (2013) 3412–3424.

[25] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, DRCW-OVO: distance-based relative competence weighting combination for one-vs-one strategy in multi–class problems, Pattern Recogn. 48 (1) (2015) 28–42.

[26] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (2) (1992) 241–260.

[27] S. Kang, S. Cho, Optimal construction of one-against-one classifier based on meta-learning, Neurocomputing 167 (2015) 459–466.

[28] E. Menahem, L. Rokach, Y. Elovici, Troika - an improved stacking schema for classification tasks, Inform. Sci. 179 (24) (2009) 4097–4122.

[29] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1991) 37–66.

[30] S. García, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Trans. Pattern Anal. Machine Intell. 34 (3) (2012) 417–435.

[31] G.M.-M. noz, D. Hernández-Lobato, A. Suárez, An analysis of ensemble pruning techniques based on ordered aggregation, IEEE Trans. Pattern Anal. Machine Intell. 31 (2) (2009) 245–259.

[32] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[33] J. Derrac, S. García, F. Herrera, Ifs-coco:instance and feature selection based on cooperative coevolution with nearest neighbor rule, Pattern Recogn. 43 (6) (2010) 2082–2105.

[34] N. García-Pedrajas, A. de Haro-García, J. Pérez-Rodríguez, A scalable approach to simultaneous evolutionary instance and feature selection, Inform. Sci. 228 (2013) 150–174.

[35] N. García-Pedrajas, J.A.R. del Castillo, D. Ortiz-Boyer, A cooperative coevolutionary algorithm for instance selection for instance-based learning, Mach. Learn. 78 (3) (2010) 381–420.

[36] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Mult.-Valued Logic Soft Comput. 17 (2011) 255–287.

[37] J.R. Quinlan, C4.5: programs for Machine Learning, 1st edition, San Mateo-California: Morgan Kaufmann Publishers, 1993.

[38] L. Rokach, Decision forest: twenty years of research, Inform. Fusion 27 (2016) 111–125.

[39] W.W. Cohen, Fast effective rule induction, in: Proceedings of Twelfth International Conference Machine Learning, ICML, 1995, pp. 1–10.

[40] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[41] S. García, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.

[42] A. Fernández, M. Calderón, E. Barrenechea, H. Bustince, F. Herrera, Solving mult-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations, Fuzzy Sets Syst. 161 (23) (2010) 3064–3080.

[43] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Aggregation schemes for binarization techniques. methods' description, Research Group on Soft Computing and Intelligent Information Systems, 2011 Tech. rep.. http://sci2s.ugr.es/ovo-ova/AggregationMethodsDescription.pdf.

[44] L.I. Kuncheva, J.C. Bezdek, R.P.W. Duin, Decision templates for multiple classifier fusion: an experimental comparison, Pattern Recogn. 34 (2) (2001) 299–314.

[45] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in: Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, pp. 211–218.

[46] A. Omari, A.R. Figueiras-Vidal, Post-aggregation of classifier ensembles, Inform. Fusion 26 (2015) 96–102.

[47] M. Kull, P.A. Flach, Reliability maps: a tool to enhance probability estimates and improve classification accuracy, in: T. Calders, F. Esposito, E. Hüllermeier, R. Meo (Eds.), Machine Learning and Knowledge Discovery in Databases, Vol. 8725 of LNSC, Springer, 2014, pp. 18–33.

[48] N. Garcia-Pedrajas, D. Ortiz-Boyer, Improving multiclass pattern recognition by the combination of two strategies, IEEE Trans. Pattern Anal. Machine Intell. 28 (6) (2006) 1001–1006.

[49] A. Arruti, I. Mendialdua, B. Sierra, E. Lazkano, E. Jauregi, New one versus all-one method: Nov@, Expert Syst. Appl. 41 (14) (2014) 6251–6260.

[50] S.-H. Park, J. Fürnkranz, Efficient prediction algorithms for binary decomposition techniques, Data Min. Knowl. Discov. 24 (1) (2012) 40–77.

[51] A. Rocha, S.K. Goldenstein, Multiclass from binary:expanding one-versus-all, one-versus-one and ecoc-based approaches, IEEE Trans. Neural Netw. Learn. Syst. 25 (2) (2014) 289–302.

[52] G. Ou, Y.L. Murphey, Multi-class pattern classification using neural networks, Pattern Recogn. 40 (1) (2007) 4–18.

[53] E.M. nés, J. Barranquero, J. Díez, J.J. del Coz, Enhancing directed binary trees for multi-class classification, Inform. Sci. 223 (2013) 42–55.

[54] J.L. Eshelman, The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination, in: G.J.E. Rawlins (Ed.), Foundations of Genetic Algorithms, San Francisco, CA: Morgan Kaufmann, 1991, pp. 265–283.

[55] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, Pattern Recogn. 46 (12) (2013) 3460–3471.

[56] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, Empowering difficult classes with a similarity-based aggregation in multi-class classification problems, Inform. Sci. 264 (2014) 135–157.

[57] C.W. Hsu, C.J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.

[58] J.C. Platt, A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans, Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, in: Advances in Large Margin Classifiers, MIT Press Cambridge, MA., 2000, pp. 61–74.

[59] J.G. Moreno-Torres, J.A. Saez, F. Herrera, Study on the impact of partition-induced dataset shift on *k*-fold cross-validation, IEEE Trans. Neural Netw. Learn. Syst. 23 (8) (2012) 1304–1312.

[60] J.L. Hodges, E.L. Lehmann, Rank methods for combination of independent experiments in analysis of variance, Ann. Math. Statist 33 (1962) 482–497.

[61] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (1979) 65–70.

[62] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bull. 1 (6) (1945) 80–83.

[63] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst. Man Cybern Part C 42 (4) (2012) 463–484.

[64] A. Fernández, V. López, M. Galar, M.J. del Jesus, F. Herrera, Analysing the classification of imbalanced data-sets with multiple classes: binarization techniques and ad-hoc approaches, Knowl.-based Syst. 42 (2013) 97–110.
[65] T.K. Ho, M. Basu, Measuring the complexity of classifier problems, in: Proceedings of 15th International Conference on Pattern Recognition (ICPR), Vol. 2, Barcelona, Spain, 2000, pp. 43–47.
[66] J.A. Sáez, J. Luengo, F. Herrera, Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification, Pattern Recogn. 46 (1) (2013) 355–364.