Pattern Recognition Letters 34 (2013) 603–609

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

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

Margin-based ordered aggregation for ensemble pruning

Li Guo [a], Samia Boukir [b,*]

[a] GAMA Laboratory, University of Lyon 1, Bt. Nautibus 43 Bld du, 11 Novembre 1918, 69622 Villeurbanne, France
[b] G&E Laboratory (EA 4592), IPB/University of Bordeaux, 1 Allée F. Daguin, 33670 Pessac, France

ARTICLE INFO

Article history:
Received 5 July 2012
Available online 11 January 2013

Communicated by S. Sarkar

Keywords:
Ensemble learning
Ensemble pruning
Margin
Ordered aggregation
Bagging

ABSTRACT

Ensemble methods have been successfully used as a classification scheme. The reduction of the complexity of this popular learning paradigm motivated the appearance of ensemble pruning algorithms. This paper presents a new ensemble pruning method which highly reduces the complexity of ensemble methods and performs better than complete bagging in terms of classification accuracy. More importantly, it is a very fast algorithm. It consists in ordering all base classifiers with respect to a new criterion which exploits an unsupervised ensemble margin. This method highlights the major influence of low margin instances on the performance of the pruning task and, more generally, the potential of low margin instances for the design of better ensembles. Comparison to both the naive approach of randomly pruning base classifiers and another ordering-based pruning algorithm is carried out in an extensive empirical analysis.

1. Introduction

Ensemble learning is a popular learning paradigm which builds a classification model by integrating multiple component learners (Dietterich, 2000).

Typically, an ensemble classifier can be built at four different levels (Kuncheva, 2004): data level (Breiman, 1996; Freund and Schapire, 1997), feature level (Ho, 1998), classifier level and combination level (Kuncheva, 2002). At data level, the modified data sets are applied to train each classifier in the ensemble, the base classifier should have a *high variance*, that is, small changes in the training set will lead to large changes in the classifier output. Neural Networks and Decision Trees are two examples of unstable classifiers. The ensemble classifier can also be built at feature level, by either disjointing or overlapping features. Different subsets of features are generated for building each classifier (Ho, 1998). At classifier level, the focus is on deciding how many and what types of classifiers to use. Finally, different ways of combining the classifier are discussed at combination level.

Bagging (Breiman, 1996) and Adaboost (Freund and Schapire, 1997) are the most widely used and successful methods at data level. Bagging is the acronym of *bootstrap* (Efron and Tibshirani, 1994) *aggregating*. It is made of the ensemble of bootstrap-inspired classifiers produced by sampling with replacement from training instances and uses these classifiers to get an aggregated classifier. The basic idea is to use different samples of the training set which are slightly different from the original set but sufficiently diverse to produce different classifiers to combine as an ensemble.

Ensemble methods have been successfully used in many fields such as bioinformatics (Pang and Zhao, 2008) and remote sensing (Guo et al., 2011). However, they have an important inconvenience: both memory required to store the parameters of the classifiers in the ensemble and processing time needed to produce a classification increase linearly with the number of classifiers in the ensemble (Martínez-Muñoz et al., 2009). In fact, it is not always true that the larger the size of an ensemble, the better it is (Zhou et al., 2002). All of these reasons motivate the appearance of ensemble pruning algorithms (Tsoumakas et al., 2009). The latter are also commonly referred to as *ensemble thinning* or *ensemble selection* algorithms. The challenge of ensemble pruning is to reduce the number of components of the ensemble while maintaining, even improving, the performance of the ensemble (Zhang et al., 2006). It is a successful strategy at classifier level.

We propose here an innovative ensemble pruning method based on the margin paradigm. The fact that it is the margin of a classification rather than the raw training error that matters has become a key tool in recent years when dealing with classifiers (Bartlett et al., 2000). Hence, the difficult examples on which focus the *boosting* are also the instances with the lowest margins (Freund and Schapire, 1997). Our pruning method is based on ordering the base classifiers, thus with a lower complexity compared to other state-of-the-art approaches (Martínez-Muñoz et al., 2009). Like in *boosting*, low margin instances play a key role in building our *pruned* ensemble classifier. However, it is not an ensemble building strategy at data level but at classifier level instead, the focus being on the selection of the right classifiers for the best ensemble.

* Corresponding author. Tel.: +33 5 57 12 10 26; fax: +33 5 57 12 10 01.
E-mail addresses: li.guo@univ-lyon1.fr (L. Guo), samia.boukir@ipb.fr (S. Boukir).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.01.003

This paper is organized as follows. The margin of ensemble methods is outlined in the following section. Section 3 briefly describes existing methods in ensemble pruning. We introduce then our ordering ensemble pruning method. The validation of our approach is presented in Section 5. Comparison to both the naive approach of randomly pruning base classifiers and another ordering-based pruning algorithm is conducted. Discussions and concluding remarks are given in Section 6.

2. Margin of ensemble methods

The margin theory of ensemble methods was originally applied to understand and evaluate ensemble methods (Schapire et al., 1998). The larger the margin, the more confidence in the classification. The smaller the margin, the closer is *a priori* the related instance to class boundaries.

In a classification problem, the instances lying on the boundaries of classes are interesting because they contain more significant information about the classes. In this case, the true class labels of these instances are not of significance. We use an unsupervised definition of the margin to emphasize these particular examples, which can be computed by Eq. (1), where c_1 is the most voted class for sample x and v_{c_1} the number of related votes, c_2 is the second most popular class and v_{c_2} the number of corresponding votes (Guo et al., 2010a; Guo, 2011). This margin's range is from 0 to +1. Furthermore, this margin concept does not require the true class label of instance x. Thus, it is potentially more robust to noise as it is not affected by errors occurring on the class label itself.

$$\text{margin}(x) = \frac{v_{c_1} - v_{c_2}}{\sum_{c=1}^{L}(v_c)} = \frac{\max_{c=1,\ldots,L}(v_c) - \max_{c=1,\ldots,L \cap c \neq c_1}(v_c)}{T} \quad (1)$$

where T represents the number of base classifiers in the ensemble.

3. Ensemble pruning

Before briefly describing main existing ensemble pruning methods, some common notation is introduced. The original ensemble of T base classifiers C_t is denoted as $C = \{C_1, C_2, \ldots, C_t, \ldots, C_T\}$. The evaluation function of a pruning method is formulated on a set of N data, which will be called *Pruning set* denoted as $V = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_N, y_N)\}$, where x_i is a vector with feature values and y_i is the value of the class label. $|y|$ is the number of classes of the classification task.

3.1. Pruning mechanism

Traditionally, as many of base classifiers as possible are created and all of them are employed to constitute an ensemble. It is true that to get a high and stable classification accuracy one requires a sufficient number of base classifiers in the ensemble. However, the more base classifiers are created, the higher the probability of getting similar base classifiers, the less diversity is produced (Kuncheva and Whitaker, 2003) and the more resources are wasted.

Ensemble pruning is a strategy which focuses on searching for a good subset of ensemble members that performs as well as, or better than, the original ensemble (Zhang et al., 2006). Besides improvement in classification performance, main benefits of ensemble pruning are lower storage requirements and higher classification speed. The complexity of finding out the best sub-ensemble of an original ensemble of size T is $O(2^T)$. It is a NP-complete problem (Garey and Johnson, 1979), whose complexity will exponentially grow with the size T growing. Furthermore, one has to construct an original ensemble of large size to achieve a better sub-ensemble. Therefore, computing the exact solution by exhaustive search is unpractical for typical ensemble sizes. Instead, the use of approximate algorithms that ensure near-optimal sub-ensembles is more appropriate.

3.2. Pruning methods

Many ensemble pruning methods have been proposed in literature and can be classified into the following categories (Tsoumakas et al., 2009):

- *Ordering*-based method. These techniques are conceptually the simplest. They firstly order the base classifiers once according to an evaluation function and then select the base classifiers in this fixed order to compose the sub-ensemble. *Kappa pruning* (Margineantu and Dietterich, 1997) is the first proposed approach for ensemble pruning. This method orders all pairs of classifiers based on the diversity measured by the κ statistic. It attempts to select the subset of most diverse classifiers. *Orientation ordering* (Martínez-Muñoz and Suárez, 2006) is another efficient method of ordering-based pruning, which orders the classifiers by increasing the value of the angle between their signature vector and a reference vector. *Accuracy ordering* orders all classifiers by their accuracy, and then aggregates the best first M classifiers to compose a sub-ensemble.
- *Clustering*-based method, which employs a clustering algorithm to discover groups of classifiers that have a low diversity, and then each cluster is separately pruned in order to increase the overall diversity of the ensemble. According to the different clustering methods involved in the pruning algorithm, some typical clustering-based pruning techniques can be mentioned such as *hierarchical agglomerative clustering pruning* (Giacinto et al., 2000), *K-means pruning* (Lazarevic and Obradovic, 2001) and *deterministic annealing* (Bakker and Heskes, 2003).
- *Optimization*-based method. Ensemble pruning is also an optimization problem: *find the subset of the original ensemble that optimizes an evaluation function*. This function is usually based on typical criteria in machine learning such as accuracy, margin or diversity. According to the optimization process involved in their model design, some of the methods of this category are referred to as semi-definite programming (Zhang et al., 2006), genetic algorithms (Zhou and Tang, 2003) and *hill climbing* (Yang et al., 2005). Due to costly non linear optimization procedures, the *optimization*-based methods are significantly more time consuming than the former approaches.

The simplest techniques are ordering-based methods. Since they are based on reordering, their computational costs (linear in the size of the initial number of base classifiers T) tend to be lower than the direct selection methods. In (Martínez-Muñoz et al., 2009), the authors have shown that *ordering-based* methods are competitive, in terms of classification accuracy, with computationally more costly methods that directly select optimal sub-ensembles.

4. A new ensemble pruning method based on the margin paradigm

4.1. Margin-based criterion

To efficiently handle classification problems, we concentrate on smaller margin instances which are closer to class boundaries and thus carry potentially more information about classes (Guo et al., 2010a,b). Furthermore, the class boundaries are the most challenging and thus are usually not well classified.

We propose an innovative margin-based criterion on choosing better classifiers. This measure involves the unsupervised ensemble margin presented before (Eq. (1)). It is defined as:

$$H(X) = -\frac{1}{|X|}\sum_{i=1}^{|X|}\log(\text{margin}(x_i)) \quad (2)$$

where x_i is an instance of set X.

This criterion is consistent with what was stated before about the significance of information that is related to the instances margin. The lower the margin of instance x_i, the higher is the information quantity, and thus the more significant is criterion $H(X)$ computed on set X.

4.2. Margin-based ordering of ensemble members

We propose here an ordering-based ensemble pruning method which relies on measuring the margin-based information quantity of each base classifier in the ensemble (Guo, 2011).

Our method is one of overproduce-and-choose (Partridge and Yates, 1996) methods which can be divided into overproduction and choose phases. In overproduction phases, adequate classifiers are constructed in large numbers, thus overproducing them. In choose phases, firstly, these overproduced classifiers, C_t, $t = 1,\ldots,T$, are used to classify a pruning set V and calculate the margin, defined in Eq. (1), of each instance in this set. Then, the margin-based criterion is computed for each classifier C_t :

$$H_t(V) = -\frac{1}{N}\sum_{i=1}^{N}\log(\text{margin}(x_i)) \quad \forall(x_i,y_i)\in V/C_t(x_i)=y_i \quad (3)$$

where x_i is an instance of set V that has been well classified ($\text{margin}(x_i) \geqslant \frac{1}{T}$) by classifier C_t. Thus, $0 \leqslant H_t(V) \leqslant \log(T)$.

All classifiers are then ordered according to this ranking measure thus leading to a sorted list of classifiers with potentially a decreasing reliability. This means that classifiers that correctly predict instances that have a low margin – i.e., the difficult instances – will be highly ranked and included in the pruned ensemble. Finally, depending on the desired amount of pruning, we choose the first M classifiers to compose a pruned ensemble classifier. Generally, M is determined through the optimization of an evaluation function, from pruning set. This function is one or some of criteria in machine learning such as overall accuracy, margin and diversity. More specific criteria, such as the accuracy of minor classes, usually the most difficult to classify, can also be involved in the evaluation function.

Finally, let us emphasize that our pruning method only depends on the votes and the number of base classifiers composing the ensemble. Thus, it is not affected by the underlying base classifiers.

4.3. Complexity

The time and space complexities of our pruning method, in terms of the initial number of base classifiers, T, and the size of the pruning set, N, are both $O(TN)$. We assume that the outputs are stored in a matrix of size $T \times N$. However, for large data sets, it might not be possible to store the whole matrix in memory, which would slow down the classification process. For an ensemble composed of T members, the ordering procedure can be carried out using the *quicksort* algorithm, which has a time complexity $O(T\log(T))$. If only the first M classifiers need to be extracted, the *quickselect* algorithm can be used. This further reduces the average running time to $O(T)$. Thus, the time required to perform the ordering is nearly linear in the number of elements of the original ensemble.

4.4. Algorithm

In this work, we used bagging to create an ensemble, involving Classification and Regression Trees (CART) as base classifier (Breiman et al., 1984). The evaluation function of best sub-ensemble selection is overall accuracy. Our algorithm involves the following steps:

1. Using bagging to generate T CART $C_1, C_2, \ldots, C_T$ from training set. Each CART was pruned by just changing the minimum size of terminal nodes.
2. Classifying the pruning set by these CART and calculating the margin of each instance in pruning set.
3. Computing the margin-based (defined by Eq. (3)) criterion $H_t(V)$ of each CART C_t, then ordering these CART according to it. Let us recall that only the well classified instances, by CART C_t, are considered in the computation of $H_m(C_t)$. The original random order of base classifiers $C_1, C_2, \ldots, C_T$ is replaced by an ordered sequence $C'_1, C'_2, \ldots, C'_T$ such that $H_m(C'_t) > H_m(C'_{t+1})$, $t \leqslant (T-1)$.
4. Choosing the M first ordered CART to compose a pruned ensemble having the best overall accuracy for pruning set. Thus, the size M of the pruned ensemble represents the number of the initial subsequence S of these CART that results in the highest overall accuracy on pruning set. The CART being already ordered, this optimization step is straightforward and very fast to compute. More formally, M is calculated as follows:

$$M = |S|$$

with:

$$S = \{C'_1,\ldots,C'_M\} = \arg\max_{t,(t\leqslant T)}\text{accuracy}$$
$$(C = \{C'_1,\ldots,C'_t\})/H_m(C'_1) > \cdots > H_m(C'_t) \quad (4)$$

5. Evaluating this sub-ensemble over test set.

This algorithm has been implemented in R project (Development Core, 2009) and led to the empirical analysis described in Section 5.

4.5. Discussion

Unlike other ordering-based methods where each classifier is independently evaluated, our algorithm uses a more global evaluation through the margin-based measure. Indeed, this criterion involves instance margin values that result from a majority voting of the whole ensemble. Thus, our pruning technique is not only based on individual properties of ensemble members (e.g., accuracy of individual learners). It also takes into account some form of complementarity of classifiers.

Our new ordering measure deliberately favors classifiers with a better performance in classifying low margin samples. Thus, it is a boosting-like strategy which aims to increase the performance on low margin instances. Therefore, our strategy of selection will lead to a subset of classifiers with a potentially improved capability to classify complex data in general, and border data in particular. Consequently, it will induce a selection of a subset of learners that are designed to efficiently handle minor classes.

At last, let us emphasize that the probability that consecutive margin-based ordering classifiers do not make coincident errors is high, because they have been ranked based on their ability to correctly classify difficult instances. Thus, this ranking based on low margin instances induce more diversity in the sequence of ordered classifiers than in other ordering-based methods, of same complexity, such as the accuracy ordering-based method. Classifiers whose errors are uncorrelated are said to be complementary. Therefore, our method tend to choose complementary classifiers. It is based on a ranking evaluation measure which integrates some diversity, but not exclusively as in *Kappa* pruning, and some accuracy but not exclusively as in *accuracy ordering* pruning.

5. Experimental results

5.1. Data sets

To evaluate our margin-based ensemble pruning scheme, we ran experiments on 10 representative data sets from the UCI repository (Asuncion and Newman, 2007), an airborne urban image, as well as a multi-source data resulting from the combination of Airborne Laser Scanning data with this color image data. Each data set has been divided into three parts: training set, pruning set and test set, as shown in Table 1. For all the data sets, the size of the original ensemble is 501.

5.2. Ensemble pruning performance comparative analysis

In this empirical analysis, our ensemble pruning method is compared to both the naive approach of randomly pruning base classifiers and another ordering-based pruning algorithm, namely the accuracy ordering-based ensemble pruning method. The latter is the most similar competing method to our approach. Indeed, accuracy ordering pruning evaluates each single classifier on all of instances. Our method focuses on single classifier's performance but just on small margin instances. Our approach will also be compared to the complete ensemble bagging approach, in which no pruning takes place. All of the results shown in the following are the mean value of a 10 time calculation.

5.2.1. Overall classification performance

Table 2 presents the average and the standard deviation of the classification accuracy obtained on *test set* by the chosen sub-ensemble that led to the maximum classification accuracy on *pruning set* for both ordering-based ensemble pruning methods. The classification accuracy of complete bagging, thus involving the full ensemble of base classifiers, is also shown for comparison. This table shows that our method outperforms the accuracy ordering pruning method, as well as using the entire ensemble for classification, in terms of classification accuracy. It is worth mentioning that our method increases the accuracy by almost 10% compared to complete bagging on data set *Tic-tac*.

5.2.2. Classification performance per class

Table 3 presents the maximum classification error rate (on average) per class obtained on *test set* by the chosen sub-ensemble of both ordering-based ensemble pruning methods as well as complete bagging. Relevant data sets are highlighted in the table. Three data sets are not of significance because the corresponding error rate is huge (>80%) probably due to noise or insufficient data for the related class. This table clearly shows that our method largely outperforms (on 8 data sets among the 9 relevant data sets) the accuracy ordering pruning method, as well as using the entire ensemble for classification, in terms of handling minor and complex classes. Indeed, they are obviously subject to the highest classification error rates. The achieved improvement is up to 5% with respect to accuracy ordering and up to **14%** compared to complete bagging (data set *Letter*). Thus, our method not only improves the overall classification accuracy compared to accuracy ordering and complete bagging (see Table 2), but also significantly reduces the maximum classification error rate per class, thus *boosting* the classification performance on difficult classes. These results demonstrate that the involvement of low margin instances in the pruning mechanism is an appealing solution for building efficient ensemble classifiers. Besides, the effectiveness of our method in handling difficult classes encourages its application to *cost-sensitive learning*.

Table 1
Data sets.

Data set	Train.	Prun.	Test	Attrib.	Classes
Connect-4	2000	2000	2000	42	3
Glass	72	71	71	9	6
Kr-vs-kp	1065	1065	1065	36	2
Letter	2000	2000	2000	16	26
Optdigits	1000	1000	1000	64	10
Pendigit	2000	2000	2000	16	10
Pima	256	256	256	8	2
Tic-tac	310	310	310	9	2
Waveform	1000	1000	1000	21	3
Wine quality-red	533	533	533	11	6
Airborne image	8000	94265	93395	3	4
Multi-source	8000	94265	93395	6	4

Table 2
Accuracy by the selected sub-ensemble of margin ordering, accuracy ordering and by complete bagging on test set.

Data set	Margin ordering (%)	Accuracy ordering (%)	Complete bagging (%)
Connect-4	73.94 ± 0.26	72.95 ± 0.57	72.65 ± 0.25
Glass	68.73 ± 2.95	67.04 ± 2.67	61.97 ± 0.13
Kr-vs-kp	99.00 ± 0.26	98.96 ± 0.26	97.06 ± 0.09
Letter	71.16 ± 0.64	68.46 ± 0.44	68.18 ± 0.33
Optdigits	95.12 ± 0.51	93.44 ± 0.42	93.59 ± 0.25
Pendigit	94.85 ± 0.17	92.96 ± 0.58	92.50 ± 0.16
Pima	72.58 ± 1.66	70.55 ± 1.72	68.55 ± 0.53
Tic-tac	84.94 ± 1.18	83.06 ± 1.22	76.29 ± 0.49
Waveform	81.32 ± 0.97	79.41 ± 0.55	80.35 ± 0.41
Wine quality-red	62.05 ± 1.28	61.83 ± 1.60	60.78 ± 0.43
Airborne image	81.63 ± 0.08	81.47 ± 0.06	81.43 ± 0.03
Multi-source	93.61 ± 0.04	93.45 ± 0.03	93.15 ± 0.01

Table 3
Maximum classification error rate (average) per class by the selected sub-ensemble of margin ordering, accuracy ordering and complete bagging, for all data sets. Relevant data sets are indicated in gray filling color.

Data set	Margin ordering (%)	Accuracy ordering (%)	Complete bagging (%)
Connect-4	99.49	99.54	98.48
Glass	90.00	88.33	88.33
Kr-vs-kp	1.24	1.28	3.31
Letter	52.68	57.80	66.67
Optdigits	11.05	13.37	12.11
Pendigit	12.59	14.57	15.85
Pima	52.18	51.72	45.98
Tic-tac	41.20	43.30	48.70
Waveform	25.05	28.02	30.77
Wine quality-red	97.50	97.00	99.50
Airborne image	66.67	67.21	67.61
Multi-source	41.15	42.08	46.14

5.2.3. Complexity

The classification speed mainly depends on the number of classifiers in the ensemble and the complexity of the base classifiers: *CART* trees in the present case. For CART trees, most of improvements in classification computational costs are expected to arise from the smaller number of classifiers in the pruned ensembles. Fig. 1 exhibits, for all data sets, the average number of classifiers involved in the selected sub-ensemble, by the two methods of ordering-based pruning, on pruning set. Our technique has a lower time complexity than accuracy ordering ensemble pruning with an average of 33 base classifiers per dataset, about **30%** less CART trees than accuracy-ordering pruning which involves 43 base classifiers on average per dataset.

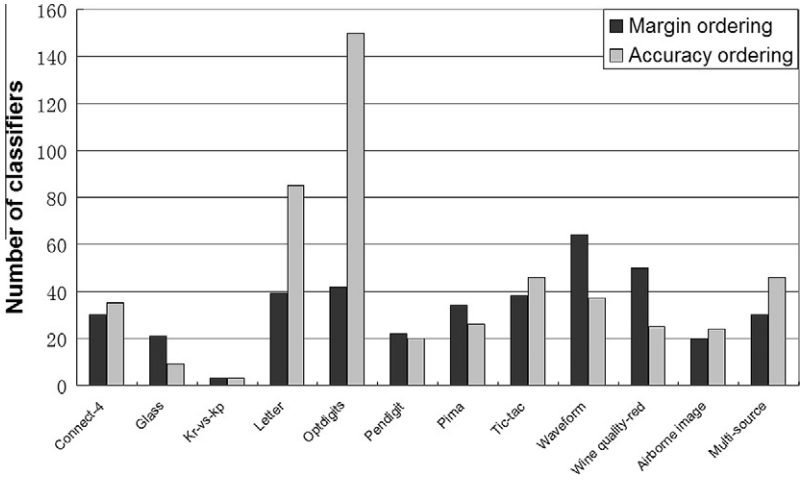

Fig. 1. Number of base classifiers selected by the ordering-based pruning methods, on pruning set, for all data sets.

Table 4
Noisy classification accuracy by the selected sub-ensemble of margin ordering, accuracy ordering and complete bagging on test set for all data sets from UCI repository.

Data set	Margin ordering (%)	Accuracy ordering (%)	Complete bagging (%)
Noise = 5%			
Connect-4	70.89 ± 0.41	70.39 ± 0.39	70.87 ± 0.18
Glass	60.84 ± 2.87	58.16 ± 2.65	59.01 ± 1.40
Kr-vs-kp	92.94 ± 0.37	92.91 ± 0.75	92.24 ± 0.11
Letter	66.87 ± 0.82	64.29 ± 0.63	63.48 ± 0.33
Optdigits	89.83 ± 0.33	88.75 ± 0.39	88.78 ± 0.20
Pendigit	89.72 ± 0.33	87.82 ± 0.48	87.50 ± 0.22
Pima	70.54 ± 0.84	69.57 ± 1.76	62.14 ± 0.56
Tic-tac	79.67 ± 1.28	76.83 ± 1.69	73.93 ± 0.75
Waveform	77.52 ± 1.06	76.73 ± 1.06	77.67 ± 0.40
Wine quality-red	59.66 ± 0.51	58.16 ± 1.28	57.42 ± 0.56
Noise = 10%			
Connect-4	67.93 ± 0.31	67.31 ± 0.56	67.31 ± 0.11
Glass	64.92 ± 3.00	64.64 ± 3.72	67.74 ± 1.81
Kr-vs-kp	86.61 ± 0.39	86.66 ± 0.67	86.15 ± 0.20
Letter	62.50 ± 0.94	60.68 ± 0.63	60.03 ± 0.31
Optdigits	83.83 ± 0.34	82.37 ± 0.69	82.83 ± 0.29
Pendigit	86.09 ± 0.28	84.06 ± 0.25	84.19 ± 0.13
Pima	67.03 ± 0.82	68.24 ± 1.48	66.17 ± 0.84
Tic-tac	72.54 ± 3.17	74.64 ± 1.02	71.38 ± 1.02
Waveform	73.20 ± 0.75	72.58 ± 0.43	73.20 ± 0.41
Wine quality-red	54.8 ± 1.37	55.17 ± 0.75	54.37 ± 0.44

5.2.4. Robustness to noise

We investigate in this section the performance of the ordering-based ensemble pruning methods in noisy classification problems. Experimental results on classification robustness of complete bagging is also reported for reference. In these experiments, classifiers are built using corrupted versions of the original data. The class labels of a fixed percentage of examples selected at random are modified. We performed two different experiments for which the class labels in the training, pruning and test sets are modified with a probability of 5% and 10%. Table 4 shows the obtained classification results on the test set of 10 data sets from the UCI repository. As in the experiments without noise, the best overall results correspond to the margin ordering ensemble pruning method. Let us notice that the poor classification results reported on data sets *Glass* and *Wine quality-red* are due to their small size compared to the number of the related classes. Hence, the impact of noise on these data sets is significantly higher than on the others.

5.2.5. Diversity

Whitaker and Kuncheva (Whitaker and Kuncheva, 2003) have shown that there is no clear or strong relationship between any of the diversity measures and the majority vote accuracy. But, they argued that the general motivation for designing diverse classifiers is correct. Non-pairwise diversity measures are calculated by counting a statistical value of all ensemble classifiers to measure the whole diversity. We use here one of the most popular non-pairwise diversity measures: *KW diversity measure* (Kohavi and Wolpert, 1996) to evaluate the diversity of the subsets of classifiers provided by both ordering-based pruning methods and random aggregation bagging. Fig. 2 shows the behavior of KW diversity with respect to the number of ranked classifiers included in the sub-ensemble by the three pruning methods on data set *Pendigit*. We can notice that our method exhibits more diversity and converges more rapidly to an asymptotic level than its counterpart pruning methods. A similar behavior has been noticed for other datasets (Guo, 2011). Though the increase in diversity with respect to bagging (random ordering) is slight, it is significant compared to accuracy ordering. Furthermore, it is well known that the random approach induces a high diversity, but without achieving the best accuracy. Thus, highlighting the role of low margins in the pruning mechanism does induce diversity in the resulting limited sub-sequence of ordered classifiers. These results confirm that our method tends to produce *diverse* classifiers as discussed in Section 4.5.

5.2.6. Boosting low margins

Fig. 3 displays a margin data histogram for well classified samples of data set *Letter* by the selected sub-ensemble of both margin ordering and accuracy ordering, as well as by complete bagging on pruning set. The histogram has been normalized. Noticeable differences can be seen between the margin distributions provided by the three *competitive* methods. The rate of the highest margin instances ($\text{margin} > 0.9$) is lower for our method compared to accuracy ordering pruning and complete bagging. In addition, the rate of the lowest margin instances ($\text{margin} < 0.3$) is lower for our method with respect to the two other techniques. By contrast, the rates of moderate margin instances ($0.3 \leqslant \text{margin} \leqslant 0.9$) are globally higher for our approach than for the others. A similar behavior has been observed for other datasets (Guo, 2011). Thus, our method exhibits a *better* dispersion of the margin values by *boosting* the lowest margins which are *shifted* towards the moderate margin values, hence the increase of the moderate margin rate, sacrificing in the process some of the highest margin instances which are also shifted towards the moderate margin values. It is

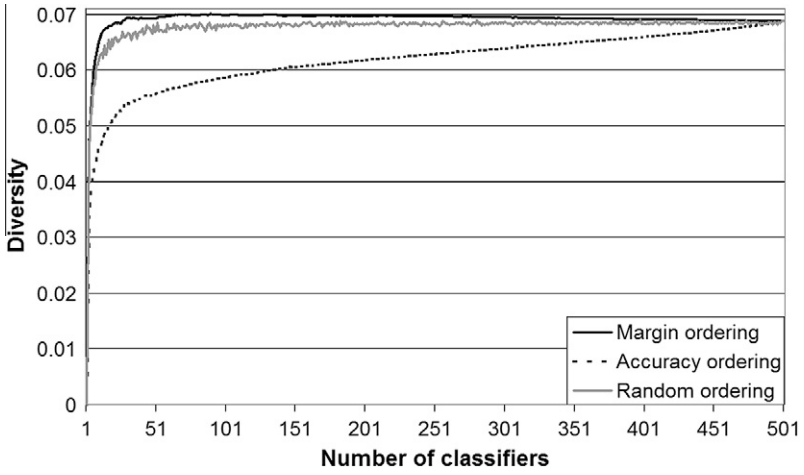

Fig. 2. Diversity of margin ordering, accuracy ordering and random ordering bagging on *Pendigit*'s test set.

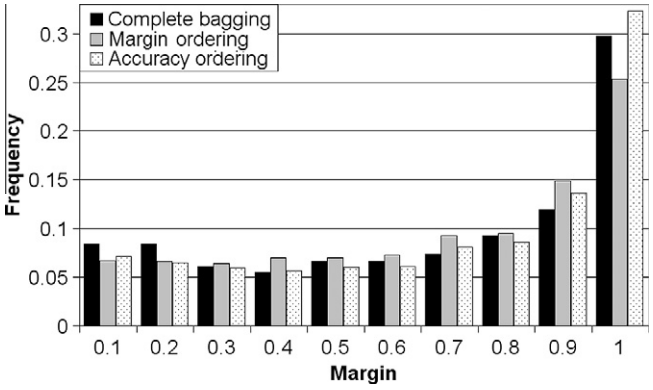

Fig. 3. Margin histogram for well classified data by the selected sub-ensemble of both margin ordering and accuracy ordering, and complete bagging on *Letter*'s pruning set.

important to notice, that the new margin dispersion induced by our pruning algorithm does not sacrifice any of the classification performance (as confirmed by our classification results shown before). On the contrary, it increases the overall classification confidence by increasing the low margin values.

6. Conclusion

We have presented a new ordering-based ensemble pruning algorithm exploiting an unsupervised version of the margin of ensemble methods. Our pruning strategy considers the smallest margin instances as the most significant in building reliable classifiers. The presented technique ranks the base classifiers according to a margin-based criterion. This method not only largely reduces the complexity of ensemble methods but also performs better than the non-pruned version in terms of classification accuracy especially in case of difficult classes. This approach also outperformed the accuracy ordering-based pruning method. Our algorithm is among the fastest methods for ensemble pruning, with a time complexity linear in ensemble size. Besides, our algorithm has a potential capability for classifying imbalanced data since it exploits low-margin samples. We are currently investigating the ability of our algorithm to efficiently tackle this challenging problem. It would also be interesting to apply our method to *cost-sensitive learning* thanks to its effectiveness in handling complex classes. Another issue is the use of a separate validation set (*pruning set*) for the evaluation of the pruning algorithms. Indeed, setting aside a validation set may induce a performance loss, especially for small or complex data sets. To overcome this possible performance loss, an appealing solution to investigate is the OOB (Out Of Bag) data approach which has been successfully used in Random Forests classifiers. When the training set for a particular learner is drawn by sampling with replacement, about one-third of the cases are left out of the sample set. These samples are called *Out of Bag (OOB)* data. Thus, the classifier ranking evaluation measure could eventually be processed using these *OOB* data instead of *wasting* a separate validation set.

Acknowledgments

The authors thank the MATIS[1] lab from IGN[2] institute for providing the remote sensing data and the related ground truth.

References

Asuncion, A., Newman, D., 2007. UCI machine learning repository. URL http://www.ics.uci.edu/~mlearn/MLRepository.html

Bakker, B., Heskes, T., 2003. Clustering ensembles of neural network models. Neural Networks 16 (2), 261–269.

Bartlett, P., Schölkopff, B., Schuurmans, D., Smola, A. (Eds.), 2000. Advances in Large Margin Classifiers, first ed. The MIT Press, Neural Information Processing.

Breiman, L., 1996. Bagging predictors. Machine Learning 24 (2), 123–140. URL http://dx.doi.org/10.1007/BF00058655

Breiman, L., Friedman, J., Olshen, R., Stone, C., 1984. Classification and Regression Trees. Wadsworth, Publisher.

Dietterich, T., 2000. Ensemble methods in machine learning. In: 1st International Workshop on Multiple Classifier Systems. pp. 1–15.

Efron, B., Tibshirani, R., 1994. An Introduction to the Bootstrap. Chapman & Hall/CRC.

Freund, Y., Schapire, R., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55 (1), 119–139.

Garey, M., Johnson, D., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, Publisher.

Giacinto, G., Roli, F., Fumera, G., 2000. Design of effective multiple classifier systems by clustering of classifiers. In: ICPR'2000, 15th International Conference on, Pattern Recognition. pp. 160–163.

Guo, L., 2011. Margin framework for ensemble classifiers. Application to remote sensing data. PhD thesis, University of Bordeaux 3, France.

Guo, L., Chehata, N., Mallet, C., Boukir, S., 2011. Relevance of airborne lidar and multispectral image data for urban scene classification using Random Forests. ISPRS Journal of Photogrammetry and Remote Sensing, Elsevier, Vol. 66 (1), 56–66.

Guo, L., Boukir, S., Chehata, N., 2010a. Support vectors selection for supervised learning using an ensemble approach. In: ICPR'2010, 20th IAPR International Conference on, Pattern Recognition, pp. 37–40.

[1] Méthodes d'Analyse pour le Traitement d'Images et la Stéréorestitution.
[2] Institut Géographique National.

Guo, L., Chehata, N., Boukir, S., 2010b. A two-pass random forests classification of airborne lidar and image data on urban scenes. In: ICIP'2010, 17th IEEE International Conference on Image Processing, pp. 26–29.

Ho, T., 1998. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8), 832–844. URL: citeseer.ist.psu.edu/ho98random.html.

Kohavi, R., Wolpert, D., 1996. Bias plus variance decomposition for zero-one loss functions. In: 13th International Conference of, Machine Learning, ICML'96. pp. 275–283.

Kuncheva, L., 2004. Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, iSBN: 0471210781.

Kuncheva, L., Whitaker, C., 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning 51, 181–207.

Kuncheva, L.I., 2002. Switching between selection and fusion in combining classifiers: An experiment. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 32 (2), 146–156.

Lazarevic, A., Obradovic, Z., 2001. Effective pruning of neural network classifiers. IEEE/INNS International Conference on Neural Networks, In, pp. 796–801.

Margineantu, D., Dietterich, T., 1997. Pruning adaptive boosting. In: ICML'1997, 14th International Conference on, Machine Learning, pp. 211–218.

Martínez-Muñoz, G., Hernández-Lobato, D., Suárez, A., 2009. An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Transactions in Pattern Analysis and Machine Intelligence 31 (2), 245–259.

Martínez-Muñoz, G., Suárez, A., 2006. Pruning in ordered bagging ensembles. In: ICML'2006, 23rd International Conference on, Machine Learning, pp. 609–616.

Pang, H., Zhao, H., 2008. Building pathway clusters from random forests classification using class votes. BMC Bioinformatics 9 (87).

Partridge, D., Yates, W.B., 1996. Engineering multiversion neural-net systems. Neural Computation 8 (4), 869–893.

R Development Core Team, 2009. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0. URL http://www.R-project.org.

Schapire, R., Freund, Y., Bartlett, P., Lee, W., 1998. Boosting the margin: a new explanation for the effectiveness of voting methods. The Annals of Statistics 26 (5), 1651–1686.

Tsoumakas, G., Partalas, I., Vlahavas, I., 2009. Applications of Supervised and Unsupervised Ensemble Methods. Vol. 245/2009. Springer, Ch. An Ensemble Pruning Primer, pp. 1–13.

Whitaker, C., Kuncheva, L., 2003. Examining the relationship between majority vote accuracy and diversity in bagging and boosting. Tech. rep., School of Informatics, University of Wales, http://www.bangor.ac.uk/ mas00a/papers/lkcwtr.pdf

Yang, Y., Korb, K., Ting, K.M., Webb, G.I., 2005. Ensemble selection for superparent-one-dependence estimators. In: AI 2005: Advances in, Artificial Intelligence, pp. 102–112.

Zhang, Y., Burer, S., Street, W., 2006. Ensemble pruning via semi-definite programming. Journal of Machine Learning Research 7, 1315–1338.

Zhou, Z., Tang, W., 2003. Selective ensemble of decision trees. In: 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular, Computing, pp. 476–483.

Zhou, Z., Wu, J., Tang, W., 2002. Ensembling neural networks: Many could be better than all. Artificial Intelligence 137 (1–2), 239–263.