Neurocomputing 160 (2015) 18–33

Contents lists available at ScienceDirect

Neurocomputing

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

Small margin ensembles can be robust to class-label noise

Maryam Sabzevari, Gonzalo Martínez-Muñoz *, Alberto Suárez

Universidad Autónoma de Madrid, Escuela Politécnica Superior, Dpto. de Ingeniería Informática, C/Francisco Tomás y Valiente, 11, 28049 Madrid, Spain

ARTICLE INFO

Article history:
Received 15 March 2014
Received in revised form
5 December 2014
Accepted 15 December 2014
Available online 11 February 2015

Keywords:
Label noise
Bagging
Small margin classifiers
Bootstrap sampling

ABSTRACT

Subsampling is used to generate bagging ensembles that are accurate and robust to class-label noise. The effect of using smaller bootstrap samples to train the base learners is to make the ensemble more diverse. As a result, the classification margins tend to decrease. In spite of having small margins, these ensembles can be robust to class-label noise. The validity of these observations is illustrated in a wide range of synthetic and real-world classification tasks. In the problems investigated, subsampling significantly outperforms standard bagging for different amounts of class-label noise. By contrast, the effectiveness of subsampling in random forest is problem dependent. In these types of ensembles the best overall accuracy is obtained when the random trees are built on bootstrap samples of the same size as the original training data. Nevertheless, subsampling becomes more effective as the amount of class-label noise increases.

1. Introduction

The success of large margin classifiers [46,33,21,20] has prompted many researchers to posit that large margins are a key feature in explaining the effectiveness of these methods. In the context of ensembles, the margin is defined as the weighted sum of votes for the correct class minus the weighted sum of votes for the most voted class other than the correct one. The effectiveness of boosting has been ascribed to the fact that it produces large margins on the training data. The margins increase as the ensemble grows because of boosting's progressive focus on instances that are difficult to classify [43]. Nonetheless, several empirical studies put in doubt the general validity of this view [9,35]. Furthermore, efforts to directly optimize the margin (or the minimum margin) have met with mixed results [40,41]. In contrast to boosting, bagging [7], random forest [11] and class-switching [10,31] ensembles do not tend to increase the classification margins. In this paper we show that subsampling can be used to generate bagging ensembles that are robust to class-label noise in spite of having small margins. By contrast, the effectiveness of subsampling in random forest is strongly problem dependent. Nevertheless, for both types of ensembles, subsampling becomes more effective as the amount of class-label noise increases.

As discussed in [53,18], class-label noise is generally more harmful for classification accuracy than noise in the feature values. Therefore, it is important to design classifiers that are robust to errors in the class labels of the training instances. The deterioration in performance caused by this type of noise is mainly due to an increase of the variance of the classifiers [36,1,39]. Bagging is robust to class-label noise because it is a variance reduction technique. As a result of its adaptive nature, boosting reduces the classification bias as well as the variance [4,48]. However, the excessive emphasis on incorrectly labeled examples makes standard boosting algorithms ill-suited for handling this type of noise. Nonetheless, it is possible to design robust versions of boosting to address this shortcoming [40,20].

A bagging ensemble is a collection of classifiers whose predictions are combined by majority voting. Each of the classifiers in the ensemble is built on a different bootstrap sample from the original training data. In standard bagging, bootstrap samples of the same size of the original training set are used to build the individual classifiers. However, this prescription need not be optimal. Several empirical studies have shown that the generalization capacity of bagging can significantly improve when smaller bootstrap samples are used [24,52,32]. Subsampling generally makes bagging more robust to label noise [42]. The key to this improvement is how smaller sampling ratios affect isolated instances. By an isolated instance we mean one that is located in a region where the majority of neighboring instances belong to a different class. Assume a sampling ratio such that the bootstrap samples used to build the individual classifiers contain less than 50% of the original training instances. This means that each instance is present in less than half of the ensemble classifiers. Therefore, the decision on the label of a given instance is dominated by classifiers trained on bootstrap samples that do not contain that particular instance [24,32]. If the instance in question is an isolated one, it is likely to receive the class label of its neighbors (i.e. the local

* Corresponding author.

http://dx.doi.org/10.1016/j.neucom.2014.12.086
0925-2312/

majority class). If the noise is uniform, most of the incorrectly labeled instances are far from the classification boundaries. They can therefore be viewed as isolated instances. In such cases, using smaller sampling ratios reduces the influence of these isolated noisy instances. Consequently, the ensemble becomes more robust.

In summary, this paper presents a comprehensive empirical assessment of the accuracy and robustness of bagging and random forest ensembles as a function of the bootstrap sampling ratio. This study extends our previous work [42] including more datasets, algorithms and experiments. In addition, we illustrate how small margin ensembles can be resilient to class-label noise.

The paper is organized as follows: Section 2 reviews previous work on label noise, focusing on classification ensembles. Section 3 is devoted to exploring the relation between margin and accuracy for different bootstrap sampling ratios and noise levels. In Section 4 we present the results of an extensive empirical evaluation of the performance of bagging and random forest ensembles built using subsampling. The experiments are carried out in a wide range of classification tasks with different amounts of class-label noise. Finally, the conclusions of this investigation are summarized in Section 5.

2. Related work

Poor data quality and contamination by noise are unavoidable in many real-world classification problems [18,53]. This has a strong potential to mislead the learning algorithms used for automatic induction from these data. Two types of noise can be present in these problems: class-label noise and polluted feature values [18,53]. Class-label noise is the consequence of incorrect manual labeling, missing information or failures in the data measuring process. Feature noise is often the result of a faulty data gathering process [18,53]. Class-label noise typically has a more pronounced misleading effect than feature noise, except when most of the feature values are corrupted [53]. Frénay and Verleysen [18] identify three types of label noise, characterized by different statistical models: The Noisy Completely at Random Model (NCAR), in which the probability of a class-label error is independent of the values of the features, the actual class of the instance and the noise rate. To simulate this type of noise the class labels of randomly selected instances are changed to a different class label, also at random. The second model is Noisy at Random (NAR). Labelling errors in this model are assumed to occur with a different probability for each class. NAR is useful to characterize tasks in which some classes are more susceptible to mislabeling than others. The third model is Noisy Not at Random (NNAR). In this case, the probability of an error depends on the actual class label and on the values of the features. This model should be used when some regions of the feature space, such as boundaries or sparse regions, are more prone to noise than others. Noise can be handled in a preprocessing step (data cleansing) or during the learning process, assuming that the algorithms used for induction from the contaminated data are robust [18].

2.1. Data cleansing

To mitigate their harmful effects, noise and outliers can be eliminated in a preprocessing step, before the selected learning algorithm is applied. For instance, it is possible to use statistical models or clustering-based methods to detect outliers. Patterns and association rules can also be used in the cleansing process [27]. An example of a pattern-based data cleansing algorithm is described in [45,44]. In this method, local SVM's are used to identify and remove instances that are suspected to be noise. For each particular training instance, k-NN is applied to locate nearby instances. A SVM is then trained on these instances to find the optimal separating hyperplane in that neighborhood. If the label predicted by this locally trained SVM does not coincide with the actual label, the instance is identified as noisy and discarded. This cleansing method has been tested on real and artificial datasets, where it showed improvements over k-NN. In [51], noisy instances are removed based on wrappers of different classification methods. In this study, the best results were obtained by removing or cleaning instances based on the prediction of a SVM built with the rest of the training data. Noisy instances are often included in the set of support vectors by a SVM classifier. Based on this observation, Fefilatyev et al. [16] propose to manually remove support vectors that are identified as noise by an expert. Then, a new SVM is built on the cleansed dataset. This process is iterated until no more support vectors are identified as noisy instances.

2.2. Robust learning algorithms

Another strategy to deal with noise is the design of robust learning algorithms. For instance, pruning is used in decision trees to reduce overfitting: the presence of noise tends to increase the size of the decision trees induced from the contaminated training data. Pruning is thus an effective way to improve the robustness of decision trees [12,13]. Another robustifying strategy is to explicitly incorporate in the learning algorithm the fact that the values of the features and the class labels can be polluted by noise. This strategy is adopted in the construction of Credal Decision Trees [28]. These types of trees are grown using the Imprecise Info-Gain Ratio (IIGR) as a splitting criterion. In this method the values of the features and class labels are approximated using probabilities and uncertainty measures.

It is also possible to adapt the algorithms used to build Support Vector Machines to improve their robustness to class-label noise. For instance, in [47] the hinge loss is replaced by a related loss function that takes into account the amount of noise in the data. With this loss function the optimization problem becomes non-convex. Heuristic optimization methods are then used to search for the global minimum of this non-convex problem. Promising results were obtained by this robust SVM in problems with asymmetric class noise (NAR model). A drawback of this method is that it is necessary to estimate the amount of noise in the data. Another robust version of SVM, called P-SVM (Probabilistic SVM) is proposed in [37] to classify magnetic resonance medical images. The P-SVM takes as inputs not only class labels but also class probability estimates. These probabilities are used to estimate the confidence on the labeling of each instance. The lower the confidence on the label, the lower the weight of that instance in the learning process. A practical limitation of this method is that one needs both qualitative (class labels) and quantitative (class posterior probabilities) information on the classes.

The problem of induction from noisy data has also been extensively addressed in the area of ensemble learning. In [2], Ali and Pazzani analyze the behavior of multiple classifier systems in the presence class-label noise. They observed that the improvements of the ensemble with respect to a single learner are generally smaller when the training data are contaminated with class-label noise. However, the reduction is not uniform and depends on the type of ensemble used.

Noise is not always harmful. In fact, noise injection is a powerful regularization mechanism that has the potential of improving the generalization capacity and robustness of prediction systems. In particular, randomization is used to build diverse ensembles that have good generalization capacity [4,38,10,15,11,34,36,31,29,17,30,49]. Furthermore, randomized ensembles, such as bagging and random forests, have been shown to be robust classifiers. By contrast, adaptive ensembles, such as boosting, are

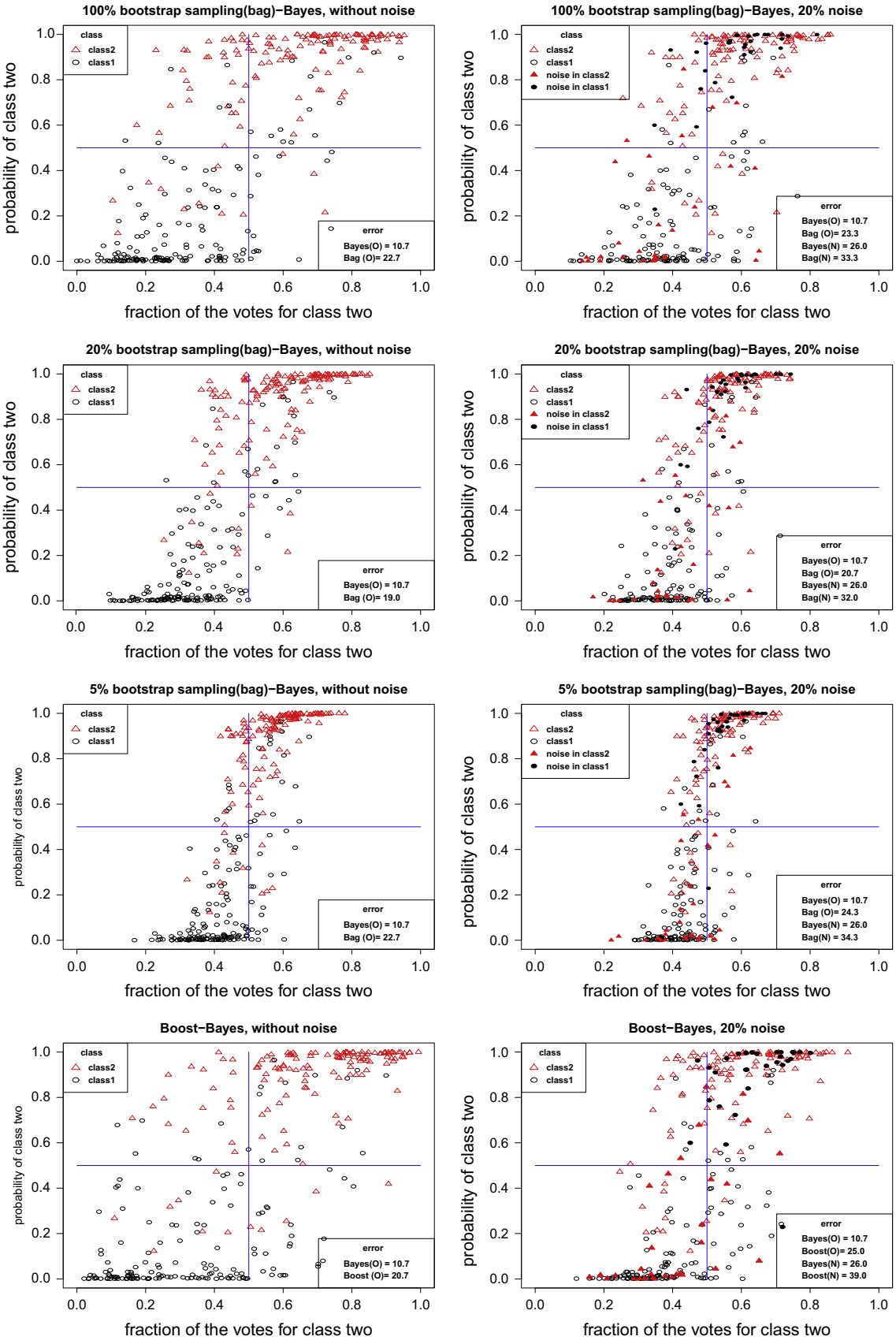

Fig. 1. Scatter plots of the posterior probability of class 2 versus the fraction of ensemble class 2 votes for each instance in the evaluation set. Results are given for *Threenorm* without noise (left column) and with 20% noise (right column). The plots correspond to bagging ensembles with sampling ratios: 100% (first row), 20% (second row) and 5% (third row). The results for boosting are presented in the forth row.

very sensitive to class-label noise [15,4,38,34,36]. The differences between these two types of ensembles can be explained by how errors are handled during the training phase: in bagging and random forest, the randomness injected during the construction of the ensemble is not correlated with the noise. For this reason, the influence of the different instances is equalized during training process [23]. By contrast, boosting increases the weights of misclassified instances irrespective of whether they are correctly labeled or not. The emphasis on correctly labeled instances that are difficult to classify is beneficial, because it reduces the classification bias. However, the focus on outliers tends to mislead the learning process. The adaptivity that makes boosting such a powerful learner also renders it overly susceptible to noise.

There are many proposals to improve the robustness of boosting to class-label noise. In most of these variants the weight update rule is modified to reduce boosting's sensitivity to noise. A successful strategy is to use less aggressive weight updates. In standard boosting the weight updates are exponential. Using

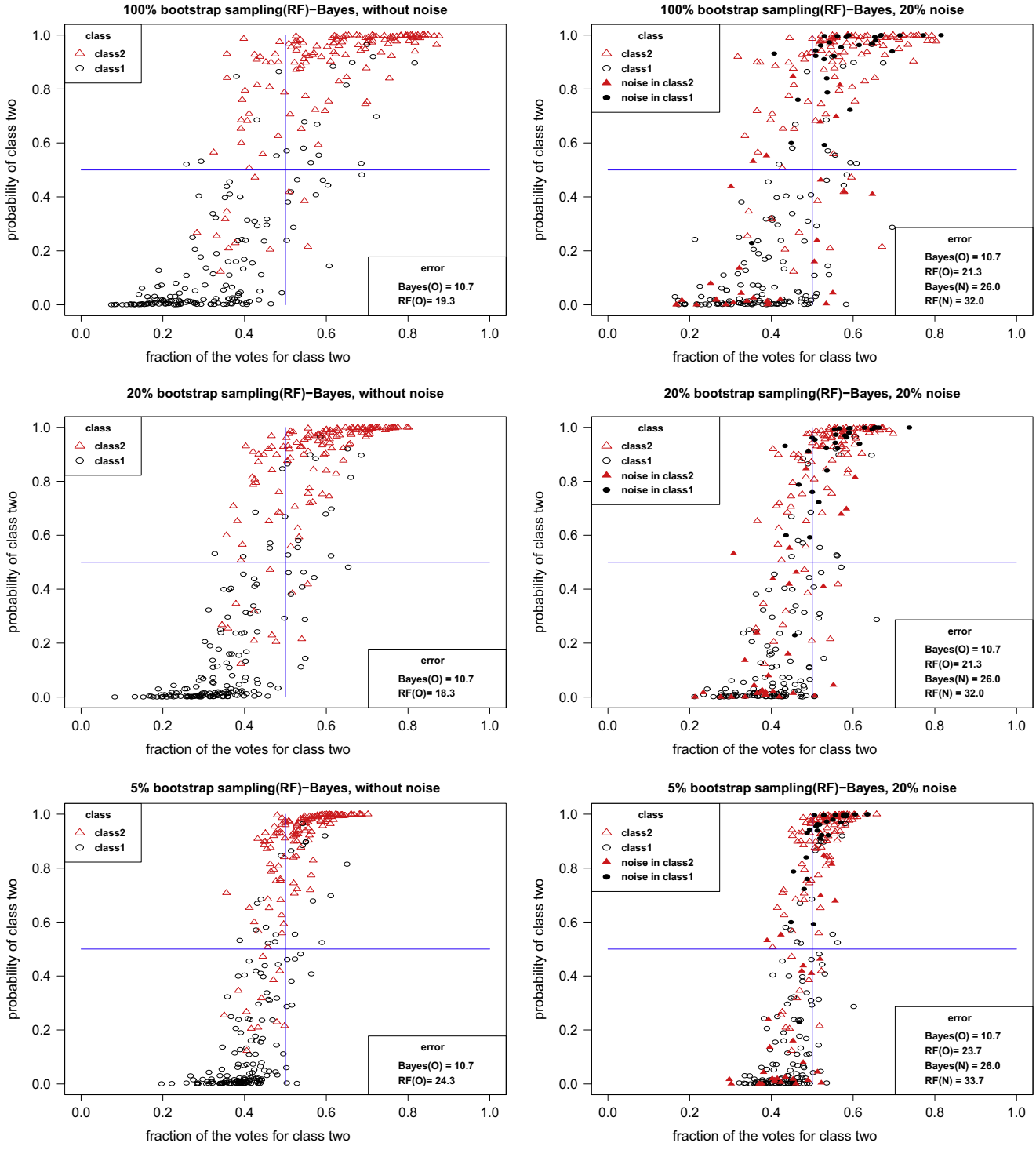

Fig. 2. Scatter plots of the posterior probability of class 2 versus the fraction of ensemble class 2 votes for each instance in the evaluation set. Results are given for *Threenorm* without noise (left column) and with 20% noise (right column). The plots correspond to random forest ensembles with sampling ratios: 100% (first row), 20% (second row) and 5% (third row).

slower updating scheme moderates the emphasis on misclassified instances. This is generally advantageous because some of this misclassified instances could be outliers [22]. In BrownBoost [19] misclassified instances with small negative margins are assigned higher weights, as in Adaboost. By contrast, instances whose margin is negative and above a specified threshold receive lower weights. The rationale behind this weight updating strategy is that instances in regions with a large class overlap tend to have low margins. By emphasizing these instances it is possible to model the classification boundary in more detail. Large negative margins correspond to isolated instances, which are far from the classification boundary. These instances are likely to be outliers and should therefore be discarded. In [34], Brownboost is shown to be more robust than Adaboost in a limited experimental setting (5 datasets for 20% class-label noise). Another way of avoiding excessive emphasis on misclassified instances is to discard instances whose weight is above a threshold [25]. The value of the threshold can be determined using a validation set. This algorithm is shown to be more robust than standard Adaboost in 8 datasets with low-medium class-label noise (up to 10%). None of these studies [34,25] compares the results of robust boosting ensembles with bagging. Finally, it is possible to combine bagging and boosting strategies to improve the accuracy and robustness of the resulting ensembles [48,26]. However, as far as we are aware, the effectiveness of these hybrid ensembles have not been systematically evaluated in experiments with class-label noise.

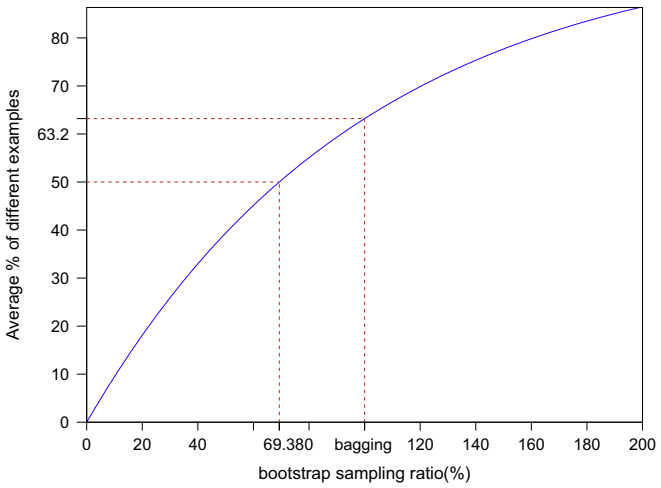

Fig. 3. Average percentage of the unique training instances with respect to the size of the bootstrap sample.

Table 1
Characteristics of the classification problems and testing method.

Dataset	Instances	Test	Attrib.	Classes
Australian	690	230	14	2
Balance	625	198	4	3
Breast W.	699	233	9	2
Diabetes	768	256	8	2
German	1000	333	20	2
Heart	270	92	13	2
Hepatitis	155	51	19	2
Horse-Colic	368	122	21	2
Ionosphere	351	117	34	2
Iris	150	50	4	3
Labor	57	38	16	2
Liver	345	115	6	2
Lung Cancer	32	10	56	3
Magic	19,020	6340	11	2
New-thyroid	215	143	5	3
Ringnorm	300	2000	20	2
Segment	2310	1540	19	7
Sonar	208	699	60	2
Threenorm	300	2000	20	2
Tic-tac-toe	958	319	9	2
Twonorm	300	5000	20	2
Vehicle	846	564	18	4
Votes	435	145	16	2
Waveform	300	5000	21	3
Wine	178	59	13	3

In [1] the authors propose to use credal decision trees to improve bagging's resilience to label noise. The results obtained with these types of ensembles in the low to medium noise regime (0–10% class-label noise) are comparable to bagging of C4.5 trees. For higher noise levels (20–30%) bagging of credal trees is more accurate than bagging of C4.5 trees.

Subsampling can also be used to design robust bootstrap ensembles. The individual classifiers of a bagging ensemble are built by applying the same base learning algorithm to different m-out-of-n bootstrap samples from the original training data. In standard bagging the number of instances in the bootstrap sample, m, is equal to the number of instances in the original training data, n (i.e. $m=n$). This choice of m need not be optimal. As an illustration, the performance of bagged nearest neighbors is comparable to the nearest neighbor algorithm itself [7]. However, if each bootstrap sample contains on average less than 50% distinct instances from the training set, the accuracy of bagged nearest neighbors can actually improve. In fact, if the sampling ratio tends to 0 as the training set size tends to ∞, the performance of bagged nearest neighbor tends to the Bayes (optimal) error [24]. Another study [52] shows that subbagging with low sampling ratios generally improves the accuracy of bagging when stable classifiers are combined. The optimal subsampling ratio can be effectively determined using out-of-bag data [32]. Subsampling has also been shown to improve the robustness of bagging to class-label noise in some classification problems [42]. In the current paper, which is an extension of this work, we present the results of a comprehensive empirical study that provide further evidence of such improvement.

A comparison of the effectiveness of these different methods cannot be done on the basis of published results. For instance, the SVM's described [47,37] are tested in very specific cases:

Table 2
Relative error change for bagging and random forest for the different levels of noise and sampling ratios. The reference value corresponds to standard bagging in the noiseless case (marked in boldface as **1.00 ± 0.00** in the table).

	Noise	10	20	40	60	80	100
Bag	0	1.38 ± 1.43	1.06 ± 0.41	0.98 ± 0.16	0.96 ± 0.08	0.98 ± 0.08	**1.00 ± 0.00**
	5	1.41 ± 1.58	1.11 ± 0.64	1.05 ± 0.27	1.08 ± 0.25	1.10 ± 0.23	1.18 ± 0.25
	10	1.45 ± 1.70	1.19 ± 0.92	1.13 ± 0.43	1.18 ± 0.44	1.28 ± 0.47	1.38 ± 0.57
	20	1.55 ± 1.98	1.42 ± 1.51	1.44 ± 1.16	1.60 ± 1.10	1.72 ± 1.13	1.83 ± 1.19
RF	0	1.77 ± 2.85	1.49 ± 2.24	1.21 ± 1.44	1.06 ± 0.91	0.97 ± 0.58	0.94 ± 0.42
	5	1.75 ± 2.62	1.48 ± 2.08	1.23 ± 1.31	1.13 ± 0.91	1.05 ± 0.68	1.01 ± 0.55
	10	1.77 ± 2.56	1.48 ± 1.98	1.27 ± 1.36	1.20 ± 1.03	1.15 ± 0.82	1.16 ± 0.76
	20	1.81 ± 2.43	1.62 ± 2.03	1.51 ± 1.55	1.48 ± 1.36	1.51 ± 1.31	1.53 ± 1.26

asymmetric noise [47] or data in which class probabilities are available [37]. An extensive empirical comparison of the different robust learning methods would be of great interest in the field. In terms of computational effort, ensembles of decision trees can be built faster than SVMs, at least in principle. Depending on the characteristics of the problem, the time complexity of SVM's is between quadratic and cubic in the number of training instances [6]. Decision trees are faster to build: their time complexity is log-

Table 3
Relative error change averaged over all datasets for bagging and random forest for the different levels of noise. The reference values are the test errors bagging and random forest noiseless case (marked in boldface in the first and fifth rows of the table).

	Noise	10	20	40	60	80	100
Bag	0	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	5	1.01 ± 0.09	1.03 ± 0.14	1.06 ± 0.14	1.12 ± 0.19	1.12 ± 0.20	1.18 ± 0.25
	10	1.03 ± 0.11	1.07 ± 0.22	1.13 ± 0.30	1.22 ± 0.38	1.30 ± 0.45	1.38 ± 0.57
	20	1.08 ± 0.13	1.23 ± 0.44	1.43 ± 0.95	1.66 ± 1.02	1.75 ± 1.12	1.83 ± 1.19
RF	0	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	5	1.05 ± 0.18	1.02 ± 0.08	1.05 ± 0.08	1.09 ± 0.16	1.08 ± 0.11	1.06 ± 0.11
	10	1.09 ± 0.30	1.06 ± 0.18	1.09 ± 0.16	1.14 ± 0.15	1.18 ± 0.18	1.21 ± 0.26
	20	1.18 ± 0.51	1.22 ± 0.39	1.35 ± 0.37	1.44 ± 0.39	1.55 ± 0.49	1.59 ± 0.53

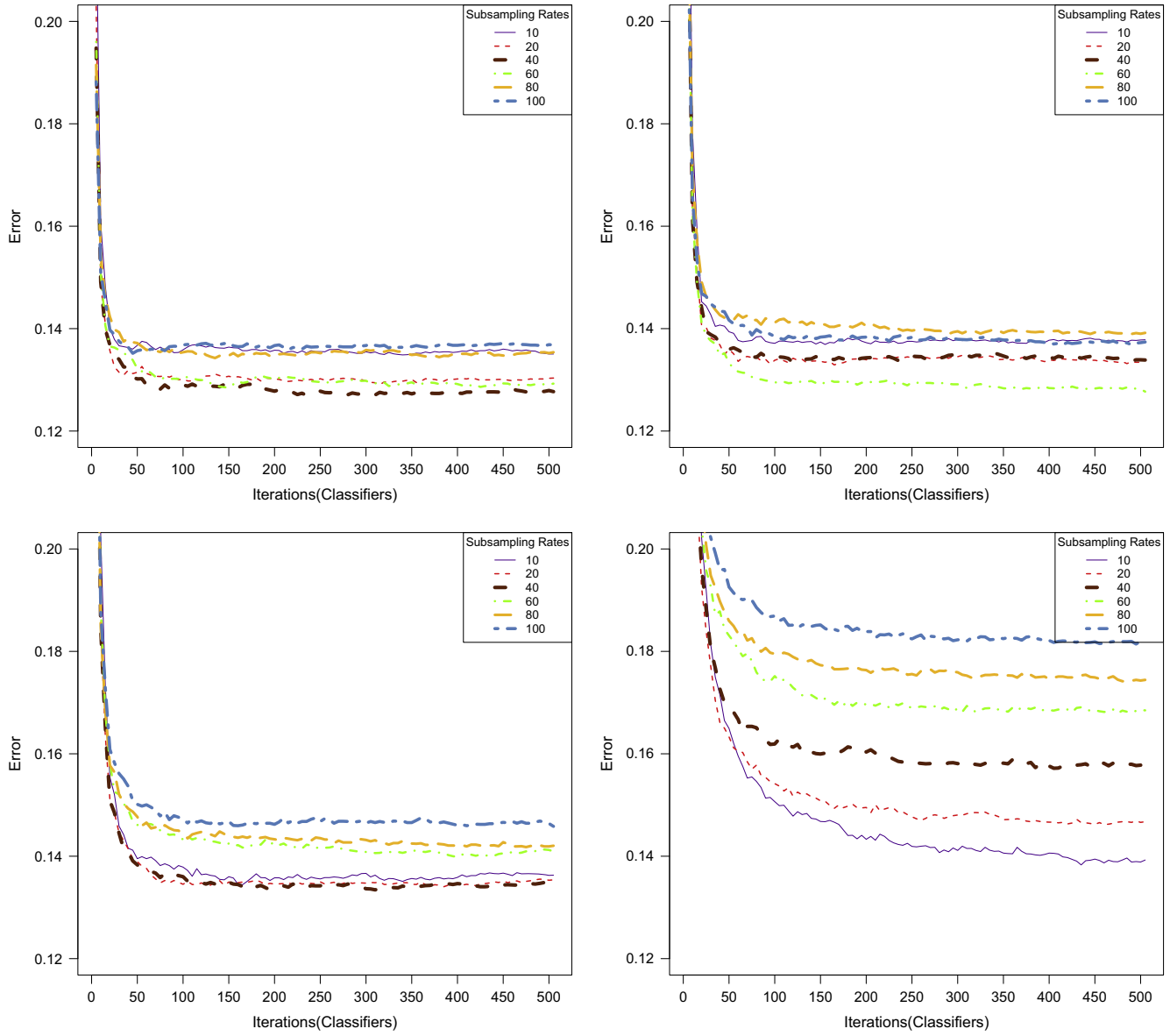

Fig. 4. Average test error of bagging in the *Australian* dataset: Noiseless setting (top left); 5% (top tight), 10% (bottom left) and 20% (bottom right) noise rates. The different curves in each plot correspond to different sampling ratios.

linear in the number of training instances and linear in the number of attributes [50]. The time needed to combine the individual decisions increases linearly with the number of base learners in the ensemble.

3. Subsampling in ensembles for noisy classification problems

In this section we explore how subsampling affects the classification margins in ensembles. The goal is to understand the relation between ensemble diversity, margins and robustness. We first present the results of a set of experiments that illustrate the effect of subsampling on the classification margin. Then we analyze how subsampling can act as a regularization mechanism that reduces the influence of mislabeled data.

3.1. Subsampling and margins

To understand how classification margins are affected by susbsampling we have carried out a series of experiments in the classification problems *Threenorm*, *Twonorm* and *Ringnorm* [9]. These are synthetic datasets for which the optimum Bayes decisions are known. Bagging ensembles and random forests of 500 trees were trained using different bootstrap sampling ratios: 100%, which is the standard prescription, 20% and 5%. Ensembles trained on a noiseless set are used as a baseline. The bagging and random forest ensembles were built on the same training sets, which consist of 300 instances. The boosting ensembles were built on different sets of the same size. Additional ensembles were then built on copies of these sets contaminated with 20% label noise. The noise was simulated using the NCAR model. Bagging and random forest ensembles were tested using the out-of-bag error [8]. The out-of-bag data of a particular classifier consists of those instances which are not included in the bootstrap sample used to build that classifier. Since they are not used for training, they can be employed as independent test data. Thus, to compute the out-of-bag error, each instance in the training set is classified using only the votes of those predictors whose training sets do not include that particular instance. Besides providing a good estimate of the generalization capacity, the out-of-bag method allows us to analyze how the injected noise is handled by the ensemble: the

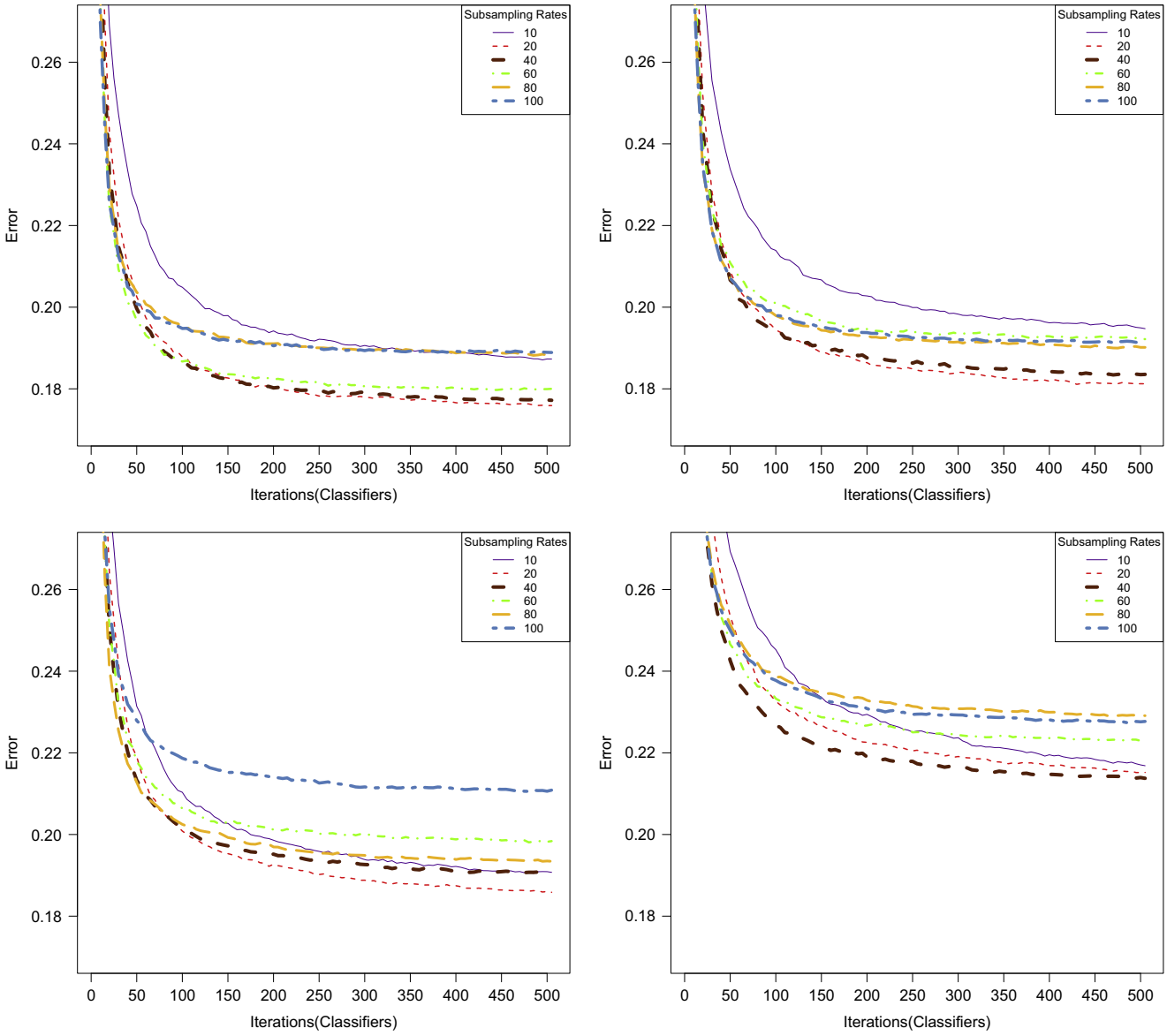

Fig. 5. Average test error of bagging in the *Threenorm* dataset: Noiseless setting (top left); 5% (top tight), 10% (bottom left) and 20% (bottom right) noise rates. The different curves in each plot correspond to different sampling ratios.

same instances, including those whose class labels have been altered, are used both for training and for testing. To allow comparisons across ensembles, the performance of boosting was evaluated on the training data used to build the bagging and random forest ensembles.

Scatter plots of the posterior probability of class 2 versus the fraction of class 2 votes for the instances in the evaluation set are given in Figs. 1 and 2. The results displayed correspond to experiments with the different ensembles, sampling ratios and class-label noise levels. In bagging and random forest, the fraction of class 2 votes for a particular instance is estimated using the classifiers for which that instance was in the out-of-bag set (i.e. the set of instances not used to train that particular classifier). For boosting, all the classifiers in the ensemble were used. Fig. 1 presents the scatter plots for an execution of *Threenorm*. Similar results are obtained in the other datasets. The plots included in this figure display (by rows) the results for standard bagging (100% sampling ratio), bagging using 20% sampling, 5% sampling and boosting. The results for a noiseless training set are presented in the first column. The results for a training set with 20% injected label noise are presented in the second column. Fig. 2 shows the corresponding plots for random forest. In all plots the class 1 (class 2) instances are marked as empty circles (triangles). The instances whose class has been changed into class 1 (class 2) are marked as filled circles (triangles). The lines shown in the plots define the decision boundaries for the Bayes classifier (horizontal line) and the ensemble (vertical line). In addition, the errors for the ensembles and the Bayes classifier are displayed on the right bottom corner of the plots. For the problems with injected label noise, error values considering noise (N) and without noise (O) are given. The Bayes classifier and the ensembles agree in the classification of instances located in the upper right and bottom left quadrants. The ensemble and the Bayes predictions are different for the remaining instances.

Several noteworthy features are revealed in these plots. In the noiseless problem (left column), the Bayes classifier assigns fairly high margins to most instances. The classification margins of bagging ensembles are lower than those of the Bayes classifier. Furthermore, they become smaller as the sampling ratio decreases. However, bagging ensembles with sampling ratios of 20% (second row) are more accurate than standard bagging, with 100% sampling (first row), in spite of the fact that the margins are smaller. The accuracy obtained with a sampling ratio of 5% is comparable to standard bagging. This is contrary to the view that accuracy should improve with increasing margin. A possible explanation of this behavior is that different bootstrap samples have fewer common instances as the sampling ratio decreases. In consequence, the base classifiers become more diverse. This increased diversity initially leads to accuracy improvements. However, if the sampling ratio is reduced beyond a threshold, the individual classifiers become inaccurate. The error reduction that results from the aggregation of their decisions in the ensemble is not sufficient to compensate the lack of accuracy of the base learners. As a result, the fraction of instances with small and negative margins increases (see 5% sampling, third row, left plot).

A similar behavior is observed when label noise is present in the training set (right column): the classification margins are now smaller in all cases, relative to the noiseless situation. The test error (second row, right column) initially improves with decreasing sampling rates. However, if the sampling ratio is too low the performance of the ensemble eventually deteriorates. A similar behavior has been reported in class-switching ensembles [31].

The behavior for boosting (last row) is somewhat different. Because of its adaptive nature, boosting produces larger margins than bagging. While this is effective in the noiseless setting, it can

Table 4
Records for statistically significant wins/draws/losses for bagging with subsampling for different sampling ratios with respect to standard bagging (100 % sampling ratio).

Noise (%)	10%	20%	40%	60%	80%
0	9/5/11	15/3/7	13/8/4	13/12/0	1/23/1
5	11/6/8	17/2/6	16/6/3	14/11/0	7/18/0
10	15/5/5	19/2/4	17/7/1	13/12/0	7/18/0
20	20/2/3	21/1/3	18/6/1	14/11/0	9/16/0

Table 5
Records for statistically significant wins/draws/losses for random forest with subsampling for different sampling ratios with respect to standard random forest (100 % sampling ratio).

Noise (%)	10%	20%	40%	60%	80%
0	1/1/23	1/3/21	2/14/9	5/12/8	3/19/3
5	0/5/20	1/6/18	2/14/9	1/19/5	1/22/2
10	3/4/18	3/9/13	3/16/6	4/19/2	3/21/1
20	8/6/11	11/5/9	9/10/6	6/16/3	3/19/3

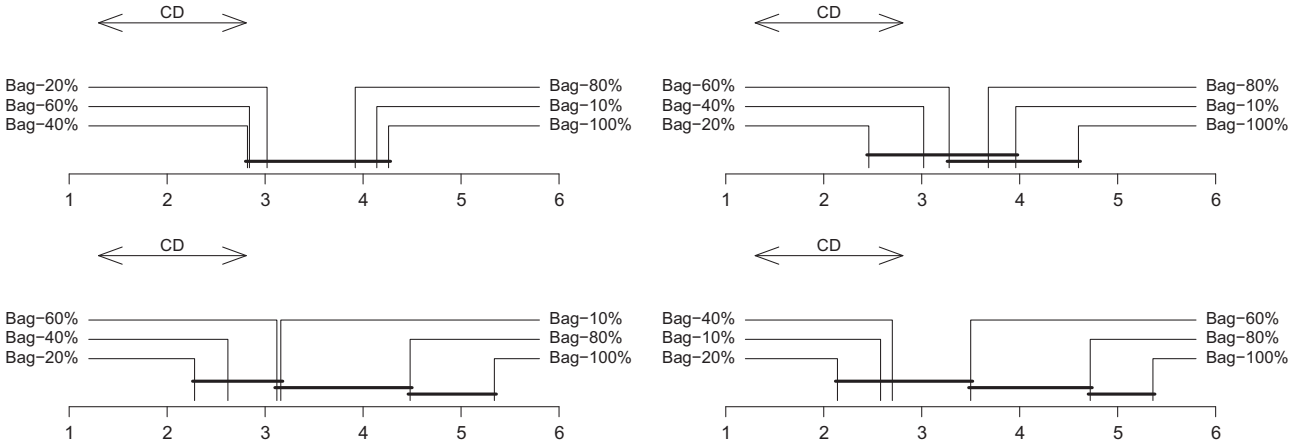

Fig. 6. Comparison of bagging with different sampling ratios using the Nemenyi test, for datasets without noise (top left) and with 5% (top right), 10% (bottom left) and 20% (bottom right) noise rates. Horizontal lines connect sampling ratios whose average ranks are not significantly different (p-value < 0.05).

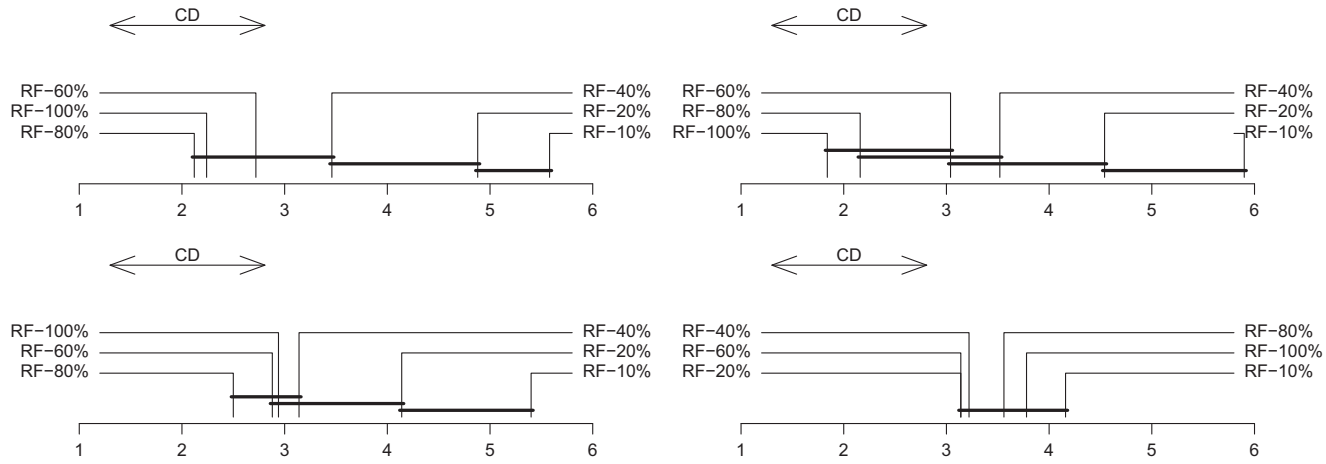

Fig. 7. Comparison of random forest with different sampling ratios using the Nemenyi test, for datasets without noise (top left) and with 5% (top right), 10% (bottom left) and 20% (bottom right) noise rates. Horizontal lines connect sampling ratios whose average ranks are not significantly different (p-value < 0.05).

Table 6
Bagging average test error I.

Dataset	Noise (%)	Bootstrap sampling ratio					
		10%	20%	40%	60%	80%	100%
Australian	0	13.5 ± 1.9	**13.0 ± 1.9**	**12.8 ± 1.6***	**12.9 ± 2.0**	13.5 ± 1.9	13.7 ± 2.2
	5	13.8 ± 1.9	13.3 ± 1.9	13.4 ± 2.0	**12.8 ± 2.0***	13.9 ± 2.0	13.7 ± 2.0
	10	**13.6 ± 1.9**	**13.5 ± 2.2***	**13.5 ± 1.9***	14.1 ± 1.7	14.2 ± 2.0	14.6 ± 2.2
	20	**13.9 ± 2.1***	**14.7 ± 1.8**	**15.8 ± 2.3**	**16.8 ± 2.6**	**17.4 ± 2.5**	18.2 ± 2.6
Balance	0	**10.2 ± 0.9***	**11.4 ± 1.7**	**13.8 ± 1.4**	**16.0 ± 1.9**	17.4 ± 1.8	18.3 ± 2.1
	5	**11.0 ± 1.1***	**11.9 ± 1.3**	**14.7 ± 1.2**	**17.1 ± 1.6**	18.3 ± 1.4	19.8 ± 2.0
	10	**11.2 ± 1.2***	**12.9 ± 1.5**	**16.3 ± 1.6**	**17.9 ± 1.4**	19.2 ± 1.9	20.3 ± 2.1
	20	**12.8 ± 1.1***	**14.9 ± 1.6**	**18.5 ± 1.8**	**20.9 ± 1.8**	23.6 ± 3.0	25.1 ± 3.2
Breast W.	0	4.1 ± 1.3	**3.7 ± 1.1***	**3.8 ± 1.0**	**3.9 ± 1.1**	4.1 ± 1.0	4.3 ± 1.1
	5	**3.7 ± 1.0**	**3.5 ± 1.0***	**3.7 ± 1.0**	**4.2 ± 1.3**	4.7 ± 1.2	5.0 ± 1.6
	10	**3.5 ± 1.2***	**3.6 ± 1.0**	**3.7 ± 1.2**	**4.4 ± 1.2**	**5.3 ± 1.5**	6.1 ± 1.7
	20	**3.5 ± 1.0***	**4.2 ± 1.3**	**5.1 ± 1.4**	**6.4 ± 1.7**	**8.0 ± 2.1**	9.2 ± 2.2
Diabetes	0	**23.7 ± 2.2***	**23.8 ± 2.1**	24.1 ± 2.6	**23.8 ± 1.9**	24.6 ± 2.3	24.4 ± 2.3
	5	**23.7 ± 2.4***	**24.2 ± 1.9**	**24.2 ± 2.3**	**24.2 ± 2.1**	24.8 ± 2.2	25.2 ± 2.3
	10	**23.4 ± 2.2***	**24.2 ± 2.4**	**24.6 ± 2.1**	**25.1 ± 2.5**	25.8 ± 2.0	25.9 ± 2.3
	20	**24.5 ± 2.3***	**24.9 ± 2.3**	**26.8 ± 2.4**	**27.1 ± 2.9**	**27.7 ± 3.0**	28.5 ± 2.5
German	0	25.0 ± 1.8	24.2 ± 1.6	23.9 ± 2.0*	23.9 ± 1.8*	24.0 ± 1.8	24.3 ± 1.8
	5	25.4 ± 1.6	**24.1 ± 1.8***	**24.3 ± 1.9**	**24.2 ± 1.7**	**24.5 ± 1.9**	25.1 ± 1.8
	10	25.5 ± 1.6	**24.6 ± 1.8***	**24.6 ± 1.8***	25.4 ± 2.1	25.8 ± 2.0	26.0 ± 2.3
	20	**26.5 ± 1.8**	**25.8 ± 1.9***	**26.4 ± 2.1**	**27.6 ± 2.5**	28.0 ± 2.5	28.5 ± 2.1
Heart	0	**17.0 ± 3.5***	**17.1 ± 3.7**	**18.7 ± 4.4**	19.0 ± 3.7	19.3 ± 3.8	19.9 ± 3.4
	5	**17.4 ± 4.0***	**18.6 ± 4.2**	**18.8 ± 4.0**	**19.3 ± 4.3**	**20.4 ± 3.7**	21.8 ± 4.6
	10	**18.1 ± 3.9***	**19.1 ± 4.6**	**19.5 ± 3.7**	21.5 ± 4.0	**21.2 ± 4.4**	22.3 ± 4.1
	20	**20.3 ± 3.8***	**22.1 ± 4.7**	**22.4 ± 4.7**	**24.3 ± 4.4**	**24.3 ± 5.1**	25.9 ± 4.3
Hepatitis	0	**21.2 ± 0.4**	**19.7 ± 2.0***	**19.8 ± 1.9**	**20.7 ± 2.7**	21.5 ± 4.0	22.2 ± 3.3
	5	**20.1 ± 2.5**	**19.8 ± 2.0***	**20.8 ± 2.6**	**21.3 ± 2.9**	**22.2 ± 3.6**	23.3 ± 4.1
	10	**20.0 ± 2.6**	**19.8 ± 1.8***	**21.1 ± 3.1**	**22.4 ± 3.7**	**23.5 ± 4.4**	25.0 ± 4.8
	20	**20.2 ± 3.6***	**20.2 ± 2.6***	**24.9 ± 4.2**	**25.8 ± 4.4**	**27.9 ± 5.6**	31.4 ± 5.2
Horse-Colic	0	25.2 ± 2.1	19.9 ± 0.9	**16.1 ± 0.4***	**16.1 ± 0.5***	17.2 ± 0.7	16.4 ± 0.9
	5	25.8 ± 2.4	21.8 ± 2.2	17.8 ± 2.1	17.1 ± 2.1	17.1 ± 1.8	17.0 ± 2.5*
	10	26.4 ± 2.5	23.3 ± 2.9	19.4 ± 2.8	18.5 ± 2.9*	18.5 ± 3.2*	18.6 ± 2.8
	20	27.5 ± 3.8	25.8 ± 3.9	22.4 ± 3.5	22.4 ± 3.8	21.3 ± 3.6*	21.9 ± 3.8
Ionosphere	0	9.6 ± 2.8	**6.8 ± 1.9***	7.5 ± 2.2	**7.2 ± 2.1**	7.7 ± 2.5	8.0 ± 2.4
	5	9.1 ± 2.5	**7.2 ± 2.3***	**7.6 ± 2.3**	8.6 ± 2.9	8.5 ± 2.6	8.4 ± 2.5
	10	9.6 ± 2.2	**7.4 ± 2.3***	**7.9 ± 2.6**	**8.1 ± 2.7**	9.1 ± 2.7	9.6 ± 2.8
	20	**10.3 ± 3.1**	**9.8 ± 3.2***	**10.1 ± 3.0**	**11.2 ± 3.5**	12.7 ± 3.4	13.0 ± 3.7
Iris	0	12.3 ± 4.6	**4.5 ± 2.6***	5.2 ± 2.7	5.3 ± 2.4	5.2 ± 2.8	5.3 ± 2.4
	5	8.6 ± 6.0	**5.1 ± 3.2***	**5.3 ± 2.8**	**5.3 ± 2.5**	7.4 ± 3.3	7.9 ± 3.7
	10	**4.6 ± 6.8***	**4.7 ± 3.4**	**5.3 ± 2.6**	**6.4 ± 3.4**	**8.9 ± 4.0**	10.6 ± 5.0
	20	**5.0 ± 3.1***	**6.1 ± 3.5**	**7.0 ± 4.6**	**10.8 ± 5.1**	**13.4 ± 5.4**	16.0 ± 6.2

be disruptive in noisy problems. In particular, when 20% class-label noise is injected boosting has the worst accuracy.

The results for random forest (shown in Fig. 2) are qualitatively similar to those of bagging. However, the margins in random forest ensemble are typically smaller than in bagging or boosting. This is a consequence of the higher diversity provided by the random trees that make up the ensemble. From the experiments performed in this study the best overall results are achieved by random forests built with the standard 100% sampling ratio. The larger initial diversity of random forest implies that there is less room for improvement as the sampling ratio decreases. The variability introduced by subsampling could in fact be detrimental to the accuracy of the ensemble. Therefore, subsampling is in general not as effective in random forest as it is in bagging. The validity of this qualitative analysis is confirmed by the empirical evidence presented in the section on experiments.

3.2. Subsampling as a regularization mechanism

Another way to understand how the sampling ratio can influence the performance of bagging ensembles is to consider the average number of distinct instances in each bootstrap sample. The dependence of this value with the sampling ratio is displayed in Fig. 3. In standard bagging (100% sampling ratio) each bootstrap sample contains on average 63.2% different instances from the original training data [8]. The remaining 36.8% are repeated instances. As the sampling ratio becomes smaller, the number of distinct instances in each bootstrap sample decreases. Eventually, only one instance is sampled for a sampling ratio of $1/N$, where N is the size of the training set. The classifier built on such a sample would predict the class label of the single instance in the sample. Hence, the ensemble decision would be the majority class in the training data, irrespective of the values of the features. On the other extreme, bootstrap samples obtained with high sampling ratios contain most of the training instances. In such cases most base learners are very similar; the diversity arises only from having different repeated examples in different bootstrap samples. Ensembles built using these extreme values of the sampling ratio will not in general have good generalization. The optimal performance is generally obtained at intermediate values of the sampling ratio [32]. Furthermore, the optimal sampling ratio need not coincide with the standard prescription (100%).

Table 7
Bagging average test error II.

Dataset	Noise (%)	Bootstrap sampling ratio					
		10%	20%	40%	60%	80%	100%
Labor	0	16.2 ± 8.8	14.7 ± 8.5	13.3 ± 9.8	11.8 ± 7.6*	13.6 ± 5.1	12.0 ± 6.4
	5	16.0 ± 10.2	13.3 ± 8.8	14.2 ± 8.8	12.4 ± 8.5	11.8 ± 7.0	10.4 ± 5.6*
	10	14.2 ± 7.2	11.8 ± 6.3*	17.6 ± 14.6	15.8 ± 6.9	17.8 ± 10.3	16.0 ± 9.4
	20	18.9 ± 8.3	17.3 ± 9.0*	17.8 ± 8.3	18.4 ± 9.6	22.9 ± 10.4	20.0 ± 10.0
Liver	0	**28.6 ± 4.0**	**27.4 ± 3.4***	**27.5 ± 3.7**	**27.8 ± 3.4**	**28.7 ± 3.6**	29.9 ± 3.6
	5	**29.0 ± 3.5**	**28.5 ± 3.8***	**28.5 ± 4.3***	**29.5 ± 4.3**	30.1 ± 3.9	30.7 ± 3.9
	10	**29.9 ± 4.0**	**29.1 ± 3.4**	**29.0 ± 4.0***	30.4 ± 3.8	31.0 ± 3.3	31.4 ± 3.7
	20	**32.4 ± 4.3**	**31.9 ± 4.5***	**32.3 ± 4.4**	**32.9 ± 4.3**	34.3 ± 4.1	34.8 ± 4.3
Lung Cancer	0	**42.0 ± 8.9***	53.5 ± 10.2	**42.0 ± 11.2***	45.5 ± 11.1	45.5 ± 11.8	45.0 ± 12.9
	5	**44.0 ± 8.5***	53.0 ± 10.8	49.0 ± 11.5	47.5 ± 12.2	46.5 ± 11.5	49.5 ± 12.7
	10	**42.0 ± 9.1***	50.5 ± 10.4	47.0 ± 11.7	45.5 ± 12.0	49.0 ± 11.9	49.0 ± 11.8
	20	**49.5 ± 9.0**	**53.5 ± 11.1**	**48.0 ± 12.0**	43.5 ± 12.8*	55.0 ± 12.2	54.0 ± 13.7
Magic	0	13.0 ± 0.4	12.5 ± 0.4	12.3 ± 0.3	12.3 ± 0.4	12.2 ± 0.4*	12.2 ± 0.4*
	5	13.1 ± 0.4	12.7 ± 0.4	12.5 ± 0.4	**12.3 ± 0.3***	12.4 ± 0.4	12.5 ± 0.3
	10	13.0 ± 0.4	12.8 ± 0.3	**12.7 ± 0.4***	**12.7 ± 0.4***	12.9 ± 0.3	12.9 ± 0.4
	20	**13.4 ± 0.4**	**13.2 ± 0.4***	**13.3 ± 0.4**	**13.6 ± 0.4**	**13.8 ± 0.4**	14.2 ± 0.4
new-thyroid	0	5.4 ± 3.0	6.4 ± 2.9	6.9 ± 3.2	5.2 ± 3.2*	5.6 ± 3.1	5.7 ± 2.7
	5	6.5 ± 2.5	**4.5 ± 1.8***	**5.3 ± 2.7**	6.7 ± 3.0	**5.0 ± 2.0**	8.3 ± 2.8
	10	6.3 ± 3.8	**5.4 ± 2.8**	5.2 ± 2.5	**5.0 ± 2.3***	6.7 ± 3.5	7.9 ± 3.6
	20	**5.2 ± 3.1**	**5.1 ± 2.7***	**5.8 ± 2.5**	10.1 ± 4.4	11.0 ± 3.6	10.6 ± 5.4
Ringnorm	0	12.1 ± 1.1	**8.1 ± 1.1**	**7.6 ± 1.3***	**8.2 ± 1.8**	8.6 ± 1.7	8.8 ± 1.9
	5	11.4 ± 1.7	**7.9 ± 1.3**	**7.4 ± 1.3***	**8.0 ± 1.7**	**8.4 ± 1.6**	9.1 ± 1.8
	10	11.3 ± 1.9	**7.8 ± 1.5**	**7.5 ± 1.5***	**8.4 ± 1.6**	**8.7 ± 1.6**	9.5 ± 1.9
	20	11.5 ± 2.1	**8.6 ± 1.5***	**9.1 ± 1.9**	**9.7 ± 1.9**	**10.1 ± 1.7**	11.2 ± 1.9
Segment	0	3.4 ± 1.4	3.0 ± 1.2	2.6 ± 1.7	2.3 ± 1.5	2.2 ± 1.0	2.1 ± 0.9*
	5	**3.2 ± 1.5**	**3.1 ± 1.3***	**3.4 ± 1.9**	3.8 ± 1.9	3.6 ± 0.7	3.8 ± 1.1
	10	**3.2 ± 1.2**	**3.1 ± 1.3***	**4.2 ± 1.1**	4.6 ± 1.7	5.2 ± 1.2	6.6 ± 1.3
	20	**3.5 ± 2.1**	**3.2 ± 1.5***	**4.0 ± 1.6**	5.7 ± 1.5	7.2 ± 1.4	7.4 ± 1.5
Sonar	0	22.5 ± 4.4	23.6 ± 4.3	23.0 ± 4.6	21.5 ± 4.9*	22.0 ± 4.7	21.0 ± 4.9
	5	24.7 ± 4.2	24.0 ± 5.3	23.2 ± 4.2	21.3 ± 4.5*	22.4 ± 4.6	22.7 ± 5.3
	10	24.8 ± 4.6	22.7 ± 5.1	23.9 ± 5.2	21.7 ± 4.8*	24.1 ± 5.4	21.8 ± 5.0
	20	**25.6 ± 5.1**	**25.7 ± 5.5**	26.8 ± 5.8	25.2 ± 5.4*	26.3 ± 5.9	26.2 ± 6.0
Threenorm	0	18.7 ± 1.2	**17.6 ± 1.3***	**17.7 ± 1.4**	**18.0 ± 1.7**	18.8 ± 1.6	18.9 ± 1.8
	5	19.5 ± 1.3	**18.1 ± 1.6***	**18.4 ± 1.4**	19.2 ± 1.8	19.0 ± 1.6	19.1 ± 1.5
	10	**19.1 ± 1.5**	**18.6 ± 1.3***	**19.1 ± 1.5**	**19.8 ± 1.5**	**19.3 ± 1.8**	21.1 ± 1.7
	20	**21.7 ± 1.9**	**21.5 ± 1.9**	**21.4 ± 1.8***	22.3 ± 1.9	22.9 ± 1.9	22.8 ± 2.0
Tic-tac-toe	0	15.4 ± 2.0	5.1 ± 2.0	2.2 ± 0.9	2.0 ± 0.9	1.9 ± 0.8*	1.9 ± 0.7 *
	5	16.9 ± 2.5	7.6 ± 2.3	3.5 ± 1.3	**3.1 ± 1.2***	3.3 ± 1.2	3.6 ± 1.4
	10	18.0 ± 2.3	10.4 ± 2.1	5.4 ± 1.7	**5.1 ± 1.6***	5.4 ± 1.6	5.6 ± 1.6
	20	20.8 ± 2.6	16.3 ± 2.7	13.0 ± 2.4	12.2 ± 2.1*	12.2 ± 2.1*	12.7 ± 2.7

An interesting regime corresponds to sampling ratios smaller than 69.3% (see Fig. 3). For values below this threshold, fewer than 50% of the original training instances are included in each bootstrap sample. This means that each instance is present in less than half of the classifiers of the ensemble. In this regime, the class label given by the ensemble for each training instance is strongly influenced by the class label of nearby instances. In consequence, subsampling has the potential to increase the diversity of the classifiers in the ensemble. Higher diversity results in more variability in the votes and therefore in lower margins. We conjecture that using sampling ratios in this regime is an effective strategy to handle class-label noise in classification ensembles.

4. Experimental evaluation

In this section we present the results of an empirical investigation of the performance of bootstrapping ensembles in the presence of label noise. The experiments are designed to assess how different sampling ratios affect the robustness of such ensembles. A total of 25 datasets from the UCI repository [3] and other sources [9] are used. They include synthetic data (*Ringnorm*, *Twonorm*, *Threenorm* and *Tic-tac-toe*) and classification problems from different application domains. The characteristics of the datasets are summarized in Table 1. They have been selected to cover a wide spectrum: there are problems with high and low numbers of attributes (e.g. *Sonar* and *Balance*, respectively), with small and large number of instances (e.g. *Magic04* and *Lung Cancer*, respectively), and with different numbers of classes.

The protocol used in the experiments is similar for all datasets. The only difference is in the generation of the training and test sets. For the synthetic datasets (*Threenorm*, *Ringnorm* and *Twonorm*) we generate a training set of 300 instances and a test set of 2000 instances. For the remaining datasets, 2/3 of the available data are used for training and 1/3 for testing. Stratified sampling is used to guarantee that the class distributions in the training and test sets are similar to the complete dataset. For each problem and realization of the training and test sets, the following steps are carried out:

1. Label noise is injected in the training set with different rates: 0% (no noise), 5%, 10% and 20%. In each case the class label of the randomly selected training instances is changed to a different class, also at random. This corresponds to the Noisy Completely At Random noise (NCAR) model [18]. Uniform noise was used to avoid making specific assumptions about the structure of the noise.
2. For each contaminated training set, six bagging ensembles composed of 500 unpruned CART (Classification And Regression Tree) trees [12] were built. The bootstrap sampling ratios used are as follows: 10%, 20%, 40%, 60%, 80% and 100% (standard bagging). The CART trees were grown until pure class nodes were obtained. No pruning was applied to the fully grown decision trees. Random forest ensembles were built on the same training sets using the different sampling ratios. Random forest is a bagging ensemble composed of random trees. In random trees the splits at the inner nodes of the tree are selected from those that involve only a subset of randomly selected features. The size of these subsets was set to the square root of the number of features for each dataset [5].
3. The generalization performance of all ensembles is gauged using the error on the test set. To obtain comparable results across all the ensembles considered no noise was injected in the test set.

The test errors reported in the tables are averages over the 100 realizations of the training and test sets.

4.1. Results

To give an overall view of the results, we have computed the averages of the test error changes in the 25 problems investigated, for each noise level, sampling strategy and ensemble method (bagging and random forest). The results are presented in Table 2 as the relative error change, using standard bagging in the noiseless setting as the reference value. This reference value is marked in boldface in the table. Values below 1 indicate that, on average, the corresponding method outperforms standard bagging in the noiseless setting. Values above 1 signal a higher average test error.

Table 8
Bagging average test error III.

Dataset	Noise (%)	Bootstrap sampling ratio					
		10%	20%	40%	60%	80%	100%
Twonorm	0	**4.9 ± 1.1**	**4.6 ± 0.8***	**5.1 ± 1.0**	**5.2 ± 0.7**	6.3 ± 1.5	6.6 ± 1.4
	5	**4.4 ± 0.7***	**5.1 ± 1.1**	5.5 ± 1.1	6.2 ± 1.9	6.2 ± 1.0	7.1 ± 2.0
	10	**5.0 ± 0.8**	**4.8 ± 0.6***	**5.9 ± 0.7**	6.6 ± 1.2	6.8 ± 1.0	7.3 ± 1.3
	20	**6.0 ± 0.5***	**7.2 ± 1.8**	7.3 ± 1.8	7.8 ± 1.1	8.4 ± 0.6	9.1 ± 1.7
Vehicle	0	26.0 ± 2.5	25.5 ± 2.3	25.5 ± 2.1	25.2 ± 2.0	25.7 ± 1.0	25.1 ± 1.1
	5	30.1 ± 2.4	28.2 ± 2.2	27.6 ± 2.0	27.4 ± 1.3	27.2 ± 1.6	26.5 ± 1.5
	10	31.8 ± 2.3	**28.4 ± 2.2**	27.9 ± 2.0	27.5 ± 1.8	28.1 ± 1.7	28.5 ± 1.2
	20	32.3 ± 2.6	29.9 ± 2.7	28.7 ± 2.5	29.0 ± 2.2	29.5 ± 2.0	29.8 ± 1.7
Votes	0	**4.4 ± 1.6**	**4.0 ± 1.6***	**4.0 ± 1.5***	**4.5 ± 1.6**	4.7 ± 1.9	5.0 ± 1.5
	5	**4.4 ± 1.4**	**4.3 ± 1.5***	**4.4 ± 1.8**	**4.5 ± 1.5**	**5.1 ± 1.7**	5.9 ± 2.1
	10	**4.5 ± 1.5***	**4.7 ± 1.5**	**4.8 ± 1.8**	**5.7 ± 1.8**	**6.7 ± 2.3**	7.3 ± 2.0
	20	**4.8 ± 1.7***	**5.8 ± 1.9**	**7.8 ± 2.9**	**9.5 ± 2.9**	**11.2 ± 3.1**	12.9 ± 3.7
Waveform	0	**17.5 ± 2.5***	**17.9 ± 2.4**	**17.8 ± 2.0**	18.8 ± 1.4	19.0 ± 1.0	20.1 ± 1.2
	5	**17.0 ± 2.6***	**17.3 ± 2.0**	**17.7 ± 1.9**	19.1 ± 1.6	19.3 ± 1.6	19.5 ± 1.5
	10	**17.5 ± 2.2***	**17.8 ± 2.2**	19.5 ± 1.6	20.8 ± 1.7	21.2 ± 1.7	21.9 ± 1.8
	20	**18.1 ± 2.7***	**19.5 ± 2.6**	19.3 ± 2.0	22.0 ± 1.5	22.2 ± 1.8	22.8 ± 1.7
Wine	0	7.6 ± 4.5	**4.5 ± 2.5**	5.2 ± 4.4	**4.4 ± 2.4**	3.9 ± 3.3*	5.1 ± 3.1
	5	6.2 ± 3.9	**3.4 ± 2.4***	**5.2 ± 2.9**	**4.9 ± 3.0**	**4.7 ± 2.9**	6.1 ± 3.6
	10	**5.9 ± 3.3**	**3.5 ± 2.2***	**4.0 ± 2.3**	**4.3 ± 3.4**	5.8 ± 4.2	7.3 ± 4.1
	20	**5.6 ± 3.4**	**4.2 ± 2.4***	**5.9 ± 3.9**	**7.0 ± 3.1**	8.7 ± 3.4	10.5 ± 4.3

In addition, the average error changes with respect to the noiseless setting for each ensemble type are shown in Table 3. The reference values are highlighted in boldface. These results serve to analyze how the accuracy of ensembles built with the different sampling ratios is affected by class-label noise. The average test error changes for the individual datasets are presented in the appendix: Tables 6–8 for bagging and Tables 9–11 for random forest ensembles.

An analysis of the results presented in Table 3 reveals that the loss of accuracy with respect to the noiseless setting is very different for different sampling ratios. For standard bagging with 20% noise injected, the average error increase with respect to the noiseless case is 83%. This large increase should be expected, given the high level of noise injected. By contrast, if a 10% sampling ratio is used, the average error increase is only 1.0%, 3.0% and 8.0% for the 5%, 10% and 20% label noise rates, respectively. An interesting observation is that these error increments are significantly lower than the corresponding levels of the noise that has been injected. Using lower sampling ratios in bagging tends to increase the variability of the base classifiers. This larger ensemble diversity generally translates into more robust classification. The remarkable robustness to class-label noise of these ensembles is illustrated in greater detail by the results presented in Tables 6–8 in the appendix. In some cases, there is even an improvement in the classification accuracy when noise is injected. For instance,the best overall accuracy of bagging in *Breast* with 20% noise is achieved using a 10% sampling ratio: the test error goes from 4.1% when no noise is injected to 3.5% when the training data has 20% noise. By contrast, when standard bagging is used, the test error increases almost 5 percentage points (from 4.3% with no noise to 9.2% with 20% noise).

For random forest ensembles, a similar, albeit less marked effect, is observed in Table 3: the deterioration with the level of noise injected is more pronounced for larger sampling ratios (18% increment with a 10% sampling ratio and 59% with a 100% sampling ratio). However, the baseline accuracy of random forest ensembles at low sampling ratios is rather poor: in the noiseless setting, the average error rate of random forest with a 10% sampling ratio is 77% larger than standard bagging (see Table 2). One of the reasons why subsampling is not as effective is that random forests are typically more diverse than bagging ensembles. This diversity makes standard random forest

Table 9
Random forest test error I.

Dataset	Noise (%)	Bootstrap sampling ratio					
		10%	20%	40%	60%	80%	100%
Australian	0	13.3 ± 1.3	6.5 ± 0.6	**4.9 ± 0.5**	**4.7 ± 0.5***	**4.9 ± 0.6**	5.1 ± 0.8
	5	14.4 ± 4.7	7.8 ± 2.2	5.7 ± 1.4	5.4 ± 1.1	5.2 ± 0.7*	5.4 ± 0.8
	10	16.6 ± 5.2	9.4 ± 3.5	6.5 ± 1.8	6.0 ± 1.3	**5.7 ± 1.0***	6.1 ± 1.1
	20	21.5 ± 6.5	13.5 ± 4.8	9.6 ± 2.5	8.9 ± 2.0	8.7 ± 2.0	8.3 ± 1.5
Balance	0	16.9 ± 2.0	15.4 ± 1.8	14.5 ± 1.9	**14.3 ± 1.6**	**14.0 ± 1.5***	15.0 ± 1.8
	5	16.1 ± 2.2	**15.2 ± 2.0**	**14.6 ± 2.0***	**14.8 ± 1.8**	**15.4 ± 1.9**	16.1 ± 2.1
	10	**15.2 ± 2.5**	**14.6 ± 1.9***	**15.7 ± 2.2**	**16.2 ± 2.1**	17.1 ± 2.2	17.6 ± 2.4
	20	**15.2 ± 2.2***	**16.0 ± 2.1**	**17.5 ± 2.6**	**19.1 ± 2.4**	19.8 ± 2.3	20.3 ± 2.9
Breast W.	0	3.5 ± 1.0	3.3 ± 1.0	3.2 ± 1.0	3.1 ± 1.0	3.0 ± 0.9*	3.0 ± 1.0*
	5	3.4 ± 1.1	3.3 ± 0.9	3.2 ± 1.0	3.2 ± 1.0	3.2 ± 0.9	3.1 ± 1.0*
	10	**3.2 ± 1.1***	**3.4 ± 1.1**	3.7 ± 1.3	3.8 ± 1.2	3.8 ± 1.1	3.8 ± 1.1
	20	**3.7 ± 1.2***	**4.2 ± 1.4**	**5.0 ± 1.5**	6.1 ± 1.9	**5.8 ± 1.5**	6.6 ± 1.7
Diabetes	0	25.8 ± 2.3	24.7 ± 2.7	24.4 ± 2.3	24.3 ± 2.3	24.2 ± 2.2	23.9 ± 2.2*
	5	25.0 ± 2.7	24.7 ± 2.7	24.6 ± 2.3	24.6 ± 2.2	24.2 ± 2.3*	24.4 ± 2.2
	10	25.2 ± 2.2	24.6 ± 2.5*	24.7 ± 2.3	25.1 ± 2.1	25.0 ± 2.3	24.7 ± 2.1
	20	**25.4 ± 2.5**	**25.3 ± 2.5***	**26.5 ± 2.5**	27.2 ± 3.0	27.0 ± 2.7	27.4 ± 2.9
German	0	29.6 ± 0.4	28.4 ± 0.7	27.0 ± 1.0	25.8 ± 1.3	25.3 ± 1.4	24.9 ± 1.3*
	5	29.2 ± 0.7	27.9 ± 1.0	26.7 ± 1.1	26.0 ± 1.2	25.3 ± 1.4	24.9 ± 1.5*
	10	28.6 ± 0.9	27.5 ± 1.3	25.8 ± 1.3	25.6 ± 1.6	25.5 ± 1.7*	25.5 ± 1.5*
	20	28.0 ± 1.3	27.2 ± 1.6	26.6 ± 1.6	26.4 ± 1.6*	27.0 ± 2.2	26.8 ± 1.9
Heart	0	20.9 ± 3.4	19.6 ± 3.9	17.5 ± 3.1	17.4 ± 3.4	17.2 ± 3.5*	17.5 ± 3.2
	5	19.8 ± 3.1	18.2 ± 3.3	17.6 ± 3.3*	18.7 ± 3.7	18.4 ± 3.3	17.7 ± 3.4
	10	19.5 ± 3.7	18.8 ± 3.7	18.5 ± 4.2*	19.1 ± 3.4	19.8 ± 3.7	18.9 ± 3.8
	20	**19.7 ± 4.3***	**20.3 ± 4.7**	**21.5 ± 3.5**	22.0 ± 4.2	22.4 ± 3.9	22.8 ± 4.9
Hepatitis	0	20.5 ± 1.1	17.9 ± 2.6	14.9 ± 3.0	13.7 ± 3.4	13.1 ± 3.3	12.7 ± 3.6*
	5	19.6 ± 2.2	15.7 ± 3.1	13.9 ± 3.3	13.0 ± 3.3	13.0 ± 3.7	12.5 ± 3.7*
	10	17.7 ± 3.5	15.7 ± 3.7	13.9 ± 3.8	13.9 ± 3.6	13.2 ± 3.4*	13.5 ± 3.9
	20	16.3 ± 4.2	15.5 ± 4.0	15.8 ± 4.2	**15.2 ± 4.4***	16.1 ± 4.4	16.5 ± 3.8
Horse-Colic	0	30.2 ± 1.7	27.6 ± 1.6	26.5 ± 1.8	26.5 ± 1.9	26.2 ± 1.8	25.3 ± 1.8*
	5	31.0 ± 3.1	27.6 ± 2.7	26.3 ± 2.2	25.8 ± 3.0	25.8 ± 2.9	24.8 ± 2.9*
	10	31.2 ± 3.4	28.3 ± 3.6	27.1 ± 3.0	25.7 ± 3.6	25.8 ± 3.3	25.6 ± 3.3*
	20	31.2 ± 4.1	29.8 ± 3.8	28.1 ± 3.6	27.4 ± 4.3	27.2 ± 3.9	26.5 ± 3.7
Ionosphere	0	12.6 ± 2.4	7.8 ± 1.9	6.6 ± 2.0	6.8 ± 1.9	6.2 ± 1.9	6.1 ± 1.8*
	5	10.3 ± 3.0	7.8 ± 2.3	7.2 ± 2.2	7.2 ± 2.3	6.8 ± 2.0*	7.1 ± 2.3
	10	10.7 ± 2.9	8.1 ± 2.2	**7.4 ± 2.3***	**7.5 ± 2.3**	7.6 ± 2.2	8.3 ± 2.7
	20	11.1 ± 3.1	**9.5 ± 2.4***	**9.6 ± 3.1**	**9.7 ± 2.6**	10.8 ± 3.2	10.6 ± 2.7
Iris	0	4.4 ± 2.5*	4.6 ± 2.2	4.4 ± 1.9*	4.7 ± 2.5	5.0 ± 2.6	4.8 ± 2.4
	5	6.2 ± 4.9	4.9 ± 3.1*	5.5 ± 2.8	5.0 ± 2.7	5.3 ± 2.9	5.4 ± 2.9
	10	7.6 ± 5.5	6.8 ± 4.5	5.6 ± 3.5*	5.7 ± 2.8	5.7 ± 3.0	6.5 ± 3.9
	20	**8.2 ± 4.8**	**7.8 ± 5.0***	**8.9 ± 5.0**	**8.9 ± 5.3**	10.6 ± 5.1	11.8 ± 5.4

more robust to noise (see rightmost column of Table 2). Using lower sampling ratios is not as effective in increasing the diversity of the random trees. Therefore, subsampling does not lead to systematic accuracy improvements in random forest ensembles.

Finally, from the analysis of the results displayed in Table 2 one concludes that the best overall performance in the noiseless setting is achieved using standard random forests (0.94). The difference with standard bagging is 6 percentage points on average. However, the difference between standard random forest and bagging using 60% sampling ratio is only of two percentage points (values 0.96 and 0.94 in Table 2). As the noise level increases the best overall accuracy corresponds to bagging using 20–40% sampling ratios (1.42 and 1.44 in the Table 2 for a 20% noise rate).

4.2. Accuracy as a function of ensemble size

The error curves displayed in Figs. 4 and 5 trace the dependence of the average test error of bagging on the number of classifiers in the ensemble. The classification problems used to illustrate this dependence are *Australian* (Fig. 4) and *Threenorm* (Fig. 5). The curves displayed correspond to different sampling ratios and noise levels: noiseless setting (top left plot), 5% (top right plot), 10% (bottom left plot) and 20% (bottom right plot) noise rates. The qualitative features of these error curves are similar in all the classification problems investigated.

When no noise is injected, the error curves for *Australian* converge to their asymptotic (infinite ensemble) limit after approximately 50 trees. As more noise is injected larger sizes are required for convergence. In this dataset the qualitative behavior of the error as a function of ensemble size is similar for the different sampling ratios. By contrast, in *Threenorm* (Fig. 5), the convergence of the ensemble error curves is slower for smaller sampling ratios.

4.3. Statistical significance of the results

A record of the statistically significant differences in accuracy with respect to the standard ensembles in the 25 classification problems investigated is given in Tables 4 and 5 for bagging and

Table 10
Random forest test error II.

Dataset	Noise (%)	Bootstrap sampling ratio					
		10%	20%	40%	60%	80%	100%
Labor	0	12.0 ± 3.7	12.7 ± 3.2	11.7 ± 2.8	11.1 ± 2.9	9.0 ± 2.8	8.9 ± 2.2*
	5	12.7 ± 3.8	12.5 ± 3.3	13.5 ± 3.6	12.5 ± 4.8	12.4 ± 3.8	11.0 ± 3.3*
	10	**12.7 ± 4.2***	**12.8 ± 4.5**	14.5 ± 4.1	14.8 ± 4.0	15.6 ± 4.5	15.7 ± 4.2
	20	**13.0 ± 5.8***	**13.5 ± 5.3**	15.5 ± 5.1	15.7 ± 4.8	16.4 ± 4.7	16.4 ± 4.5
Liver	0	36.8 ± 2.0	33.5 ± 2.2	29.7 ± 3.0	28.1 ± 2.9	27.5 ± 3.2	27.1 ± 3.2*
	5	35.1 ± 3.2	32.7 ± 3.0	29.9 ± 3.2	29.2 ± 3.5	28.8 ± 3.6	28.5 ± 4.0*
	10	33.6 ± 2.9	31.4 ± 3.9	30.8 ± 4.2	30.4 ± 3.6	30.3 ± 3.7*	30.6 ± 3.4
	20	33.9 ± 3.8	33.2 ± 4.2*	33.7 ± 4.4	34.3 ± 4.7	33.5 ± 4.4	34.4 ± 4.6
Lung Cancer	0	57.9 ± 9.1	53.8 ± 11.7	48.2 ± 12.9	**43.0 ± 13.1***	46.8 ± 13.2	48.4 ± 14.6
	5	60.7 ± 7.4	55.8 ± 11.7	49.2 ± 13.0	49.3 ± 12.3	47.6 ± 13.7	47.4 ± 12.6*
	10	61.8 ± 9.3	55.9 ± 11.7	54.2 ± 12.9	50.2 ± 15.7*	51.2 ± 13.5	51.8 ± 12.7
	20	61.8 ± 10.9	58.7 ± 12.1	55.4 ± 11.5	54.7 ± 13.0	**50.5 ± 13.3***	54.8 ± 14.2
Magic	0	14.4 ± 0.4	13.6 ± 0.4	12.9 ± 0.3	12.6 ± 0.4	12.4 ± 0.3*	12.4 ± 0.4*
	5	13.5 ± 0.4	13.2 ± 0.4	12.8 ± 0.4	12.7 ± 0.4	12.5 ± 0.3*	12.5 ± 0.3*
	10	13.3 ± 0.4	13.0 ± 0.4	12.9 ± 0.4	12.8 ± 0.4	12.7 ± 0.4*	12.7 ± 0.4*
	20	**13.6 ± 0.4**	**13.5 ± 0.4***	**13.5 ± 0.4***	**13.6 ± 0.4**	13.7 ± 0.4	13.8 ± 0.4
New-thyroid	0	8.4 ± 2.3	7.3 ± 2.5	5.1 ± 2.0	3.3 ± 1.8	3.0 ± 1.0	4.4 ± 1.2
	5	8.1 ± 2.4	8.2 ± 2.5	5.6 ± 2.3	5.8 ± 1.9	4.0 ± 1.6	3.4 ± 1.5
	10	8.2 ± 2.8	5.9 ± 2.1	3.3 ± 2.4	4.8 ± 1.8	4.3 ± 1.7	3.2 ± 1.7
	20	**6.1 ± 2.5***	**6.2 ± 3.0**	7.2 ± 2.8	8.0 ± 2.5	8.4 ± 2.7	8.5 ± 2.6
Ringnorm	0	13.2 ± 1.4	6.5 ± 0.7	4.9 ± 0.5	**4.8 ± 0.6***	**4.8 ± 0.6***	5.0 ± 0.7
	5	14.9 ± 4.7	7.4 ± 2.3	5.5 ± 1.2	5.2 ± 0.9*	5.2 ± 0.9*	5.4 ± 0.8
	10	16.7 ± 5.2	9.3 ± 3.1	6.2 ± 1.4	6.1 ± 1.2	6.0 ± 1.3*	6.1 ± 1.4
	20	21.4 ± 5.7	14.3 ± 4.8	9.7 ± 2.4	8.6 ± 2.3	8.5 ± 1.7	8.2 ± 1.8
Segment	0	5.9 ± 0.9	4.4 ± 0.9	3.5 ± 0.5	2.9 ± 0.6	2.7 ± 0.7	2.6 ± 0.6*
	5	5.9 ± 0.8	4.5 ± 0.9	3.7 ± 0.7	3.1 ± 0.8*	3.1 ± 0.7*	3.2 ± 0.8
	10	5.8 ± 1.1	4.6 ± 1.1	**3.4 ± 0.6**	**3.0 ± 0.7***	**3.4 ± 0.7**	4.1 ± 0.7
	20	5.7 ± 0.7	**4.9 ± 0.8**	**4.0 ± 0.9***	**4.5 ± 0.8**	**5.2 ± 0.7**	6.1 ± 0.7
Sonar	0	31.1 ± 4.8	24.6 ± 4.9	21.5 ± 4.6	19.6 ± 4.5	18.3 ± 4.6*	18.6 ± 4.4
	5	28.6 ± 6.1	24.9 ± 5.2	21.3 ± 5.0	20.6 ± 5.2	20.5 ± 4.5	19.9 ± 3.8*
	10	28.7 ± 6.5	24.4 ± 5.0	21.2 ± 4.5	20.7 ± 4.5	20.9 ± 5.2	20.6 ± 4.5*
	20	27.8 ± 6.4	26.3 ± 5.2	24.9 ± 4.7	24.4 ± 5.5	24.2 ± 5.9	23.9 ± 5.2*
Threenorm	0	18.2 ± 0.9	16.9 ± 0.9	16.0 ± 0.9*	16.8 ± 1.0	16.2 ± 1.0	16.0 ± 1.1*
	5	22.3 ± 3.2	19.3 ± 1.9	18.4 ± 1.2	17.2 ± 1.1	17.2 ± 1.1	16.9 ± 1.0*
	10	24.7 ± 3.9	21.7 ± 2.5	20.0 ± 1.6	19.5 ± 1.5	**18.6 ± 1.6***	19.0 ± 1.5
	20	30.6 ± 4.5	26.3 ± 3.2	23.4 ± 2.0	22.9 ± 2.2	22.3 ± 2.0	21.6 ± 1.7*
Tic-tac-toe	0	29.0 ± 1.3	23.0 ± 1.4	15.2 ± 1.7	10.0 ± 2.0	6.6 ± 2.0	4.9 ± 1.7*
	5	26.9 ± 1.9	21.5 ± 1.9	14.0 ± 1.9	10.1 ± 2.3	7.7 ± 2.0	6.3 ± 1.9*
	10	26.3 ± 2.2	20.6 ± 2.2	14.5 ± 2.4	11.3 ± 2.4	9.1 ± 2.1	8.4 ± 2.3*
	20	25.2 ± 2.5	21.3 ± 2.7	16.8 ± 2.6	14.9 ± 2.5	14.3 ± 2.4	13.6 ± 2.4*

random forest, respectively. In each cell of these tables the number of times a given method wins, draws or looses against standard bagging (Table 4) or standard random forest (Table 5) is displayed. Paired t-tests with $\alpha = 0.05$ are used to determine the significant wins and losses. A draw is recorded if the differences between the test errors are not statistically significant.

From the results presented in these tables one concludes that subsampling is more effective at higher levels of label noise. For instance, from Table 4, bagging using a 10% sampling ratio and 0% noise significantly outperforms standard bagging in 9 datasets and obtains lower accuracy in 11 datasets. When the noise rate is increased to 20%, the situation reverses: there are 20 wins and only 3 significant losses.

An analysis of the results for random forest in Table 5 leads to similar conclusions. Subsampling becomes more effective also at lower sampling ratios. The effect, however, is less salient than in bagging. In the noiseless case random forest using 20% bootstrap sampling outperforms the standard version in only one dataset and losses in 21 datasets. When the noise rate is increased to 20% the number of wins increases to 11 and the number of losses decreases to 9. Random forests built using the standard prescription (100% sampling ratio) have the best overall performance in the problems investigated for all noise levels. However, as the amount of class-label noise increases, subsampling becomes more effective and is actually advantageous in some problems.

Finally, the method proposed by Demšar in [14] is used to compare the performance of the ensembles across the different datasets. The comparison is made in terms of the average rank of each classifier in the problems considered. For a given dataset, the rank of the different ensembles is computed on the basis of the average test errors in different realizations of the training and test sets. Fig. 6 present the results of these tests for different noise levels and sampling ratios. A Nemenyi test with p-value < 0.05 is used to determine the statistical significance of the differences between average ranks. The critical distance above which these differences are considered significant is shown for reference (CD = 1.5 for 6 methods, 25 dataset and p-value < 0.05). In this diagrams, if two methods are connected with a horizontal solid line, the difference between their average ranks is not statistically significant.

Fig. 6 displays the results of the Demšar test for bagging ensembles in the noiseless setting (top left), and with 5% (top right), 10% (bottom left) and 20% (bottom right) noise rates. In all cases, standard bagging with 100% sampling ratio has the lowest average rank. When no noise is injected the highest accuracy corresponds to bagging with 20%, 40% and 60% sampling ratios. However, the differences with other sampling ratios are not statistically significant. The improvements over standard bagging for 20% and 40% sampling ratios are statistically significant in the problems with noise rates 5%, 10% and 20%. For the 20% noise rate, bagging ensembles that use 10%, 20%, 40% and 60% sampling ratios are significantly better than standard bagging (100% sampling ratio).

The results of the Demšar test for random forest are displayed in Fig. 7. Standard random forest (i.e. with 100% sampling ratio) is the best method for the noiseless setting (top left plot) and for 5% noise rate (top right plot). However, the differences with ensembles built with 80%, 60% and 40% sampling ratios are not statistically significant. For these noise rates standard random forest significantly outperform ensembles built using 20% and 10% sampling ratios. When higher noise levels are injected (10%), the best performance corresponds to random forest with 80% sampling ratio. The improvements over ensembles built with 10% and 20% sampling ratios are statistically significant. For the highest noise level (20%) the method with the highest average rank is random forest with a 20% sampling ratio. However, in this case, none of the differences between the average ranks of the different ensembles are statistically significant.

5. Conclusion

In this paper we have analyzed the resilience to class-label noise of bootstrap aggregation ensembles as a function of the size of the bootstrap samples used to train the individual predictors. The results of an extensive empirical evaluation show that bagging composed of unpruned decision trees trained on bootstrap samples whose size is between 10% and 40% of the size of the original training set are more resilient to label noise than standard bagging (i.e. using a 100 % sampling ratio). For random forests subsampling

Table 11
Random forest test error III.

Dataset	Noise (%)	Bootstrap sampling ratio					
		10%	20%	40%	60%	80%	100%
Twonorm	0	**3.3 ± 0.3***	**3.3 ± 0.3***	**3.3 ± 0.3***	**3.4 ± 0.3**	3.6 ± 0.3	3.6 ± 0.4
	5	4.5 ± 1.3	3.9 ± 0.8*	**3.9 ± 0.5***	4.0 ± 0.5	3.9 ± 0.5*	4.0 ± 0.4
	10	5.9 ± 2.0	4.8 ± 1.2	4.4 ± 0.7*	4.6 ± 0.9	4.4 ± 0.6*	4.5 ± 0.6
	20	9.5 ± 4.4	7.4 ± 2.7	6.3 ± 1.4	6.0 ± 1.2*	6.1 ± 1.0	6.3 ± 1.1
Vehicle	0	30.7 ± 2.4	29.6 ± 1.7	27.2 ± 1.9	26.3 ± 1.7	25.9 ± 1.7*	26.1 ± 1.6
	5	30.9 ± 3.0	29.0 ± 2.0	27.3 ± 2.0	26.1 ± 1.8	26.2 ± 2.1	25.6 ± 2.5*
	10	30.1 ± 2.2	27.8 ± 2.2	27.9 ± 2.4	26.2 ± 1.7	26.3 ± 2.0	25.9 ± 1.8*
	20	30.5 ± 2.2	29.6 ± 2.4	28.2 ± 2.2	27.0 ± 2.5	27.4 ± 1.8	26.9 ± 2.6*
Votes	0	5.3 ± 1.7	4.4 ± 1.5	3.9 ± 1.4	3.6 ± 1.3*	3.6 ± 1.4*	3.6 ± 1.3*
	5	5.4 ± 1.7	4.4 ± 1.5	4.0 ± 1.4	3.9 ± 1.4	3.7 ± 1.5*	3.7 ± 1.7*
	10	5.7 ± 1.6	5.0 ± 1.7	4.5 ± 1.5	**4.1 ± 1.5***	4.2 ± 2.0	4.6 ± 1.9
	20	6.3 ± 2.2	**5.5 ± 2.1***	**5.9 ± 2.2**	6.2 ± 2.3	6.5 ± 2.6	6.7 ± 2.5
Waveform	0	15.5 ± 0.7	14.9 ± 0.8	14.8 ± 0.8	14.5 ± 0.6*	14.6 ± 0.6	14.6 ± 0.6
	5	15.3 ± 0.9	15.1 ± 0.9	14.9 ± 1.1	15.0 ± 0.8	14.8 ± 0.8*	14.8 ± 0.6*
	10	15.1 ± 0.6	14.8 ± 0.5	14.8 ± 0.8	14.9 ± 0.8	14.6 ± 0.9*	15.0 ± 0.7
	20	14.9 ± 1.0*	15.0 ± 0.6	15.3 ± 0.8	15.4 ± 0.7	15.4 ± 1.0	14.9 ± 0.7*
Wine	0	3.0 ± 1.8	3.1 ± 1.9	2.5 ± 1.7	2.3 ± 1.6	2.1 ± 1.7	1.9 ± 1.6*
	5	4.8 ± 3.3	3.6 ± 2.7	2.7 ± 2.0	2.9 ± 2.2	2.8 ± 2.1	2.4 ± 2.0*
	10	5.8 ± 4.3	4.1 ± 3.0	3.7 ± 2.6	3.4 ± 2.6	3.2 ± 2.4*	3.4 ± 2.6
	20	7.0 ± 4.4	5.8 ± 3.6	5.9 ± 3.5	5.1 ± 3.1	5.3 ± 3.5	5.0 ± 3.0*

is effective only in noisy domains ($\approx 20\%$ noise in the class labels) and in specific classification tasks. In most problems, for low noise levels the best results are obtained using high sampling ratios. In fact, using the standard sampling ratio to build random forests is a reasonable choice with a good overall performance in the problems investigated, especially in the absence of class-label noise. However, in noisy tasks, it is worth to explore the possibility of subsampling, because the optimal size of the bootstrap samples is problem dependent.

Experiments in synthetic data have been used to illustrate that the classification margins become smaller as the sampling ratio decreases. This effect occurs both in the noiseless setting and when class-label noise is injected. They provide empirical evidence that using smaller bootstrap samples to build the individual ensemble classifiers can lead to improvements in accuracy, especially in noisy domains. However, if the sampling ratio decreases beyond a threshold the accuracy of the ensemble abruptly drops. This abrupt deterioration of performance occurs at higher sampling rates in random forests than in bagging. The reason is that the margins are typically larger in bagging than in random forests. Since lower sampling ratios tend to reduce the margin, the potential improvements of subsampling for random forest are realized only in problems with high levels of class-label noise.

Acknowledgements

The authors acknowledge financial support from Spanish Plan Nacional I+D+i Grant TIN2013-42351-P and from Comunidad de Madrid Grant S2013/ICE-2845 CASI-CAM-CM.

Appendix A

Tables 6–8 display the average test error (with the standard deviation after the $\pm$ sign) of bagging for the different sampling ratios and noise rates considered. The results are presented in three separate tables for the sake of clarity. In each row the lowest error is highlighted with an asterisk (*). For each noise level and dataset (i.e. for each row), values that are significantly better than standard bagging (column 100%) are highlighted in boldface. Results that are significantly worse than standard bagging are underlined. The statistical significance of these differences is determined using paired t-tests at a significance level $\alpha = 0.05$. The corresponding results for random forest ensembles are presented in Tables 9–11.

References

[1] J. Abellán, A.R. Masegosa, Bagging decision trees on data sets with classification noise, in: Foundations of Information and Knowledge Systems, Springer, Berlin, Heidelberg, 2010, pp. 248–265.
[2] K.M. Ali, M.J. Pazzani, Error reduction through learning multiple descriptions, Mach. Learn. 24 (3) (1996) 173–202.
[3] K. Bache, M. Lichman, UCI Machine Learning Repository, URL http://archive.ics.uci.edu/ml, 2013.
[4] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: bagging, boosting, and variants, Mach. Learn. 36 (1–2) (1999) 105–139.
[5] S. Bernard, L. Heutte, S. Adam, Influence of hyperparameters on random forest accuracy, in: Proceedings of the 8th International Workshop on Multiple Classifier Systems, MCS 2009, Reykjavik, Iceland, June 10–12, 2009, pp. 171–180.
[6] L. Bottou, C.-J. Lin, Support vector machine solvers, Large Scale Kernel Mach. 23 (2007) 301–320.
[7] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[8] L. Breiman, Out-of-Bag Estimation, Technical Report, Department of Statistics, University of California, 1996.
[9] L. Breiman, Arcing classifiers, Ann. Stat. 26 (1998) 801–823.
[10] L. Breiman, Randomizing outputs to increase prediction accuracy, Mach. Learn. 40 (3) (2000) 229–242.
[11] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[12] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees. Wadsworth, Monterey, CA, 1984.
[13] C.E. Brodley, M.A. Friedl, Identifying Mislabeled Training Data, CoRR abs/1106.0219, 2011.
[14] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[15] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Mach. Learn. 40 (2) (2000) 139–157.
[16] S. Fefilatyev, M. Shreve, K. Kramer, L.O. Hall, D.B. Goldgof, R. Kasturi, K. Daly, A. Remsen, H. Bunke, Label-noise reduction with support vector machines, in: ICPR, 2012, pp. 3504–3508.
[17] E. Frank, B. Pfahringer, Improving on bagging with input smearing, in: PAKDD, 2006, pp. 97–106.
[18] B. Frénay, M. Verleysen, Classification in the presence of label noise: a survey, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2014) 845–869.
[19] Y. Freund, An adaptive version of the boost by majority algorithm, in: Proceedings of the Twelfth Annual Conference on Computational Learning Theory, 2000, pp. 102–113.
[20] Y. Freund, A More Robust Boosting Algorithm, arXiv preprint, arXiv:0905.2138, 2009.
[21] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Ann. Stat. 28 (1998) 2000.
[22] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Ann. Stat. 28 (1998) 2000.
[23] Y. Grandvalet, Bagging equalizes influence, Mach. Learn. 55 (3) (2004) 251–270.
[24] P. Hall, R.J. Samworth, Properties of bagged nearest neighbour classifiers, J. R. Stat. Soc.: Ser. B (Stat. Methodol.) 67 (3) (2005) 363–379.
[25] A. Karmaker, S. Kwek, A boosting approach to remove class label noise, Int. J. Hybrid Intell. Syst. 3 (3) (2006) 169–177.
[26] A. Krieger, C. Long, A. Wyner, Boosting noisy data, in: ICML, 2001, pp. 274–281.
[27] J.I. Maletic, A. Marcus, Data cleansing: beyond integrity analysis, in: Proceedings of the Conference on Information Quality, Citeseer, 2000, pp. 200–209.
[28] C.J. Mantas, J. Abellán, Credal decision trees to classify noisy data sets, in: HAIS, 2014, pp. 689–696.
[29] G. Martínez-Muñoz, A. Sánchez-Martínez, D. Hernández-Lobato, A. Suárez, Building ensembles of neural networks with class-switching, in: Artificial Neural Networks, ICANN 2006, Springer, Berlin, Heidelberg, 2006, pp. 178–187.
[30] G. Martínez-Muñoz, A. Sánchez-Martínez, D. Hernández-Lobato, A. Suárez, Class-switching neural network ensembles, Neurocomputing 71 (13) (2008) 2521–2528.
[31] G. Martínez-Muñoz, A. Suárez, Switching class labels to generate classification ensembles, Pattern Recognit. 38 (10) (2005) 1483–1494.
[32] G. Martínez-Muñoz, A. Suárez, Out-of-bag estimation of the optimal sample size in bagging, Pattern Recognit. 43 (1) (2010) 143–152.
[33] L. Mason, P. Bartlett, J. Baxter, Direct optimization of margins improves generalization in combined classifiers, Adv. Neural Inf. Process. Syst. (1999) 288–294.
[34] R.A. McDonald, D.J. Hand, I.A. Eckley, An empirical comparison of three boosting algorithms on real data sets with artificial class noise, in: Multiple Classifier Systems, Springer, Berlin, Heidelberg, 2003, pp. 35–44.
[35] D. Mease, A. Wyner, Evidence contrary to the statistical view of boosting, J. Mach. Learn. Res. 9 (Jun. 2008) 131–156.
[36] P. Melville, N. Shah, L. Mihalkova, R.J. Mooney, Experiments on ensembles with missing and noisy data, in: Proceedings of the Workshop on Multi Classifier Systems, Springer-Verlag, Berlin, Heidelberg, 2004, pp. 293–302.
[37] E. Niaf, R. Flamary, O. Rouvière, C. Lartizien, S. Canu, Kernel-based learning from both qualitative and quantitative labels: application to prostate cancer diagnosis based on multiparametric MR imaging, IEEE Trans. Image Process. 23 (3) (2014) 979–991.
[38] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, J. Artif. Intell. Res. 11 (1999) 169–198.
[39] C. Pölitz, R. Schenkel, Robust ranking models using noisy feedback, in: Workshop "Information Retrieval Over Query Sessions" (SIR 2012), ECIR, 2012, pp. 1–6.
[40] G. Rätsch, Robust Boosting via Convex Optimization: Theory and Applications (Doctoral Thesis), 2001.
[41] G. Rätsch, M.K. Warmuth, Maximizing the margin with boosting, in: J. Kivinen, R.H. Sloan (Eds.), COLT, Lecture Notes in Computer Science, vol. 2375, Springer, Berlin, Heidelberg, 2002, pp. 334–350.
[42] M. Sabzevari, G. Martínez-Muñoz, A. Suárez, Improving the robustness of bagging with reduced sampling size. In: Proceedings of the European Symposium on Artificial Neural Networks, ESANN 2014, pp. 667–682.
[43] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. 26 (5) (1998) 1651–1686.
[44] N. Segata, E. Blanzieri, P. Cunningham, A scalable noise reduction technique for large case-based systems. In: Case-Based Reasoning Research and Development, Springer, Berlin, Heidelberg, 2009, pp. 328–342.
[45] N. Segata, E. Blanzieri, S.J. Delany, P. Cunningham, Noise reduction for instance-based learning with a local maximal margin approach, J. Intell. Inf. Syst. 35 (2) (2010) 301–331.
[46] Advances in Large Margin Classifiers, in: A.J. Smola, P.J. Bartlett (Eds.), MIT Press, Cambridge, MA, USA, 2000.
[47] G. Stempfel, L. Ralaivola, Learning SVMs from sloppily labeled data, in: ICANN, vol. 1, 2009, pp. 884–893.

[48] G.I. Webb, Multiboosting: a technique for combining boosting and wagging, Mach. Learn. 40 (2) (Aug. 2000) 159–196.
[49] D.P. Williams, Label alteration to improve underwater mine classification, IEEE Geosci. Remote Sens. Lett. 8 (3) (2011) 488–492.
[50] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd edition, Morgan Kaufmann, Burlington, MA, 2011.
[51] J. Young, J. Ashburner, S. Ourselin, Wrapper methods to correct mislabelled training data, in: Proceedings of the 2013 International Workshop on Pattern Recognition in Neuroimaging, PRNI '13, IEEE Computer Society, Washington, DC, USA, 2013, pp. 170–173.
[52] F. Zaman, H. Hirose, Effect of subsampling rate on subbagging and related ensembles of stable classifiers. In: Pattern Recognition and Machine Intelligence, Springer, Berlin, Heidelberg, 2009, pp. 44–49.
[53] X. Zhu, X. Wu, Class noise vs. attribute noise: a quantitative study, Artif. Intell. Rev. 22 (3) (2004) 177–210.

Maryam Sabzevari received her B.Sc. (2008) and M.Sc. (2010) in computer science from Azad university of Mashhad, Iran. Currently she is conducting her Ph.D. in Computer Science in Universidad Autónoma de Madrid (UAM), Madrid, Spain. Her interest include machine learning, pattern recognition, decision trees and ensemble learning.

Gonzalo Martínez-Muñoz received the university degree in Physics (1995) and Ph.D. degree in Computer Science (2006) from the Universidad Autónoma de Madrid (UAM). From 1996 to 2002 he worked in industry. Until 2008 he was an interim assistant professor in the Computer Science Department of the UAM. During 2008/2009, he worked as a Fulbright postdoc researcher at Oregon State University in the group of Professor Thomas G. Dietterich. He is currently a professor at Computer Science Department at UAM. His research interests include machine learning, computer vision, pattern recognition, neural networks, decision trees, and ensemble learning.

Alberto Suárez received the degree of Licenciado in Chemistry from the Universidad Autónoma de Madrid, Spain, in 1988, and the Ph.D. degree in Physical Chemistry from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 1993. After holding postdoctoral positions at Stanford University (USA), at the Université Libre de Bruxelles (Belgium), as a research fellow financed by the European Commission within the program "Training and Mobility of Researchers", and at the Katholieke Universiteit Leuven (Belgium), he is currently a professor in the Computer Science Department of the Universidad Autónoma de Madrid (Spain). He has also held appointments as "senior visiting scientist" at the International Computer Science Institute (Berkeley, CA) and at MIT (Cambridge, MA). He has worked on relaxation theory in condensed media, stochastic and thermodynamic theories of nonequilibrium systems, lattice-gas automata, and automatic induction from data. His current research interests include machine learning, quantitative and computational finance, time series analysis and information processing in the presence of noise.