Rotation Forest: A New Classifier Ensemble Method

Juan J. Rodríguez, *Member*, *IEEE Computer Society*,
Ludmila I. Kuncheva, *Member*, *IEEE*, and Carlos J. Alonso

Abstract—We propose a method for generating classifier ensembles based on feature extraction. To create the training data for a base classifier, the feature set is randomly split into K subsets (K is a parameter of the algorithm) and Principal Component Analysis (PCA) is applied to each subset. All principal components are retained in order to preserve the variability information in the data. Thus, K axis rotations take place to form the new features for a base classifier. The idea of the rotation approach is to encourage simultaneously individual accuracy and diversity within the ensemble. Diversity is promoted through the feature extraction for each base classifier. Decision trees were chosen here because they are sensitive to rotation of the feature axes, hence the name "forest." Accuracy is sought by keeping all principal components and also using the whole data set to train each base classifier. Using WEKA, we examined the Rotation Forest ensemble on a random selection of 33 benchmark data sets from the UCI repository and compared it with Bagging, AdaBoost, and Random Forest. The results were favorable to Rotation Forest and prompted an investigation into diversity-accuracy landscape of the ensemble models. Diversity-error diagrams revealed that Rotation Forest ensembles construct individual classifiers which are more accurate than these in AdaBoost and Random Forest, and more diverse than these in Bagging, sometimes more accurate as well.

Index Terms—Classifier ensembles, AdaBoost, bagging, random forest, feature extraction, PCA, kappa-error diagrams.

1 Introduction

Classifier combination is now an active area of research in Machine Learning and Pattern Recognition [31], [33], [34], [35], [36], [46]. Many studies have been published, both theoretical and empirical, which demonstrate the advantages of the combination paradigm over the individual classifier models [21]. A great deal of research has gone into designing multiple classifier systems based on the same classifier model trained on different data subsets or feature subsets. Such multiple classifier systems are commonly called *classifier ensembles*.[1]

Two approaches for constructing classifier ensembles seem to be perceived as "classic" at present. They have been found to be accurate, computationally feasible across various data domains, and with no clear dominance between them.

- *Bagging* [5] takes bootstrap samples of objects[2] and trains a classifier on each sample. The classifier votes are combined by majority voting. In some implementations, classifiers produce estimates of the posterior probabilities for the classes. These probabilities are averaged across the classifiers and the most probable class is assigned, called "average" or "mean" aggregation of the outputs. Bagging with average aggregation is implemented in WEKA and used in the experiments in this paper. Since each individual classifier is trained on a bootstrap sample, the data distribution seen during training is similar to the original distribution. Thus, the individual classifiers in a bagging ensemble have relatively high classification accuracy. The only factor encouraging diversity between these classifiers is the proportion of different objects in the training samples. Although the classifier models used in Bagging are sensitive to small changes in data, the bootstrap sampling appears to lead to ensembles of low diversity compared to other ensemble creating methods. As a result, Bagging requires larger ensemble sizes to perform well. To enforce diversity, a version of Bagging called *Random Forest* was proposed by Breiman [7]. The ensemble consists of decision trees built again on bootstrap samples. The difference lies in the construction of the decision tree. The feature to split a node is selected as the best feature among a set of M randomly chosen features, where M is a parameter of the algorithm. This small alteration appeared to be a winning heuristic in that diversity was introduced without much compromising the accuracy of the individual classifiers. The *Random Subspaces* method [17] builds each ensemble member on a different subset of features randomly selected form the original feature set. A common feature of all methods in this class is that the

1. While still unifying terminology across the field, in a wider sense, any system of more than one classifier can be called a classifier ensemble.
2. We use "objects" as synonym of instances, examples, and data points.

- *J.J. Rodríguez is with the Escuela Politécnica Superior, Edificio C, Universidad de Burgos, c/ Francisco de Vitoria s/n, 09006 Burgos, Spain. E-mail: jjrodriguez@ubu.es.*
- *L.I. Kuncheva is with the School of Informatics, University of Wales, Bangor, Gwynedd, LL57 1UT, UK. E-mail: l.i.kuncheva@bangor.ac.uk.*
- *C.J. Alonso is with the Departamento de Informática, Escuela Técnica Superior de Ingeniería Informática (E.T.S.I.I.), Campus Miguel Delibes s/n, Universidad de Valladolid, 47011 Valladolid, Spain. E-mail: calonso@infor.uva.es.*

Manuscript received 21 Sept. 2005; revised 3 Feb. 2006; accepted 22 Feb. 2006; published online 11 Aug. 2006.
Recommended for acceptance by M. Figueiredo.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0513-0905.

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.
0162-8828/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

individual classifiers can be built in parallel, independently of one another. This does not mean that their *outputs* will be independent though [16].

- *Boosting* [12] is a family of methods, the most prominent member of which is *AdaBoost*. The idea is to boost the performance of a "weak" classifier by using it within an ensemble structure. The classifiers in the ensemble are added one at a time so that each subsequent classifier is trained on data which have been "hard" for the previous ensemble members. A set of weights is maintained across the objects in the data set so that objects that have been difficult to classify acquire more weight, forcing subsequent classifiers to focus on them. The ensemble construction through AdaBoost comes from theory and is, in fact, equivalent to fitting an additive logistic regression model by a stage-wise estimation procedure [13], [37], [38]. The function being optimized is closely related to the classification error. Bounds on the training and generalisation errors of AdaBoost have been proven [12], [37], [39], thereby inspiring further developments of theory and practice of boosting [1], [6], [12], [25], [27], [28], [30], [40], [45].

Comparative studies can be found in [2], [3], [8], [18]. It appears that, on average, AdaBoost is the best method although Random Subspaces and Bagging have their application niches as well. Interestingly, for large ensemble sizes (in the order of thousand classifiers) the significant differences between the ensemble models almost disappear [2]. This raises a quest for a consistently good ensemble strategy for small ensemble sizes. Regardless of the computational effort required for training, small ensembles will have the advantage of fast and, in many cases, near-optimal performance [26].

Diversity is an important property of a classifier ensemble [23]. The success of AdaBoost has been explained, among others, with its diversity creating ability. Margineantu and Dietterich [26] devise the so-called "kappa-error" diagrams to show the effect of making the classifiers diverse at the expense of reduced individual accuracy. Plotted in these diagrams are $L(L-1)/2$ points, where L is the ensemble size. Each point corresponds to a pair of classifiers. On the x-axis is a measure of diversity between the pair; kappa was chosen in [26]. On the y-axis is the averaged individual error of the classifiers in the pair. Plotting a diagram for an ensemble designed by Bagging and another designed by AdaBoost made the differences between the two approaches very clear. AdaBoost was creating inaccurate classifiers by forcing them to concentrate on difficult objects and ignore the rest of the data. This however, led to large diversity which boosted the ensemble performance, often beyond that of Bagging. This leads us to the famous accuracy-diversity dilemma. It seems that classifiers cannot be both very accurate and have very diverse outputs.

In this study, we propose an ensemble construction method, called *Rotation Forests* which aims at building accurate *and* diverse classifiers. The main heuristic consists in applying feature extraction to subsets of features and reconstructing a full feature set for each classifier in the ensemble. We have chosen Principal Component Analysis (PCA) in this study for reasons explained in Section 2. The rest of the paper is organized as follows: Section 2 explains the Rotation Forests. Section 3 presents our experimental study comparing Rotation Forest with Bagging, AdaBoost and Random Forest. In Section 4, kappa-error diagrams are plotted to illustrate the pattern of relationship between diversity and individual accuracy for all ensemble methods studied here. Section 5 offers our conclusions and outlines directions of future work.

2 Rotation Forests

Let $\mathbf{x} = [x_1, \ldots, x_n]^\top$ be a data point described by n features and let X be the data set containing the training objects in a form of an $N \times n$ matrix. Let Y be a vector with class labels for the data, $Y = [y_1, \ldots, y_N]^\top$, where y_j takes a value from the set of class labels $\{\omega_1, \ldots, \omega_c\}$. Denote by $D_1, \ldots, D_L$ the classifiers in the ensemble and by $\mathbf{F}$, the feature set. As with most ensemble methods, we need to pick L in advance. All classifiers can be trained in parallel, which is also the case with Bagging and Random Forests. To construct the training set for classifier D_i, we carry out the following steps:

1. Split $\mathbf{F}$ randomly into K subsets (K is a parameter of the algorithm). The subsets may be disjoint or intersecting. To maximize the chance for high diversity, we chose disjoint subsets. For simplicity, suppose that K is a factor of n so that each feature subset contains $M = n/K$ features.
2. Denote by $\mathbf{F}_{i,j}$ the jth subset of features for the training set of classifier D_i. For every such subset, select randomly a nonempty subset of classes and then draw a bootstrap sample of objects, of size 75 percent of the data count. Run PCA using only the M features in $\mathbf{F}_{i,j}$ and the selected subset of X. Store the coefficients of the principal components, $\mathbf{a}_{i,j}^{(1)}, \ldots, \mathbf{a}_{i,j}^{(M_j)}$, each of size $M \times 1$. Note that it is possible that some of the eigenvalues are zero, therefore, we may not have all M vectors and, so, $M_j \leq M$. Running PCA on a subset of classes instead on the whole set is done in a bid to avoid identical coefficients if the same feature subset is chosen for different classifiers.
3. Organize the obtained vectors with coefficients in a sparse "rotation" matrix R_i

$$R_i = \begin{bmatrix} \mathbf{a}_{i,1}^{(1)}, \mathbf{a}_{i,1}^{(2)}, \ldots, \mathbf{a}_{i,1}^{(M_1)}, & [\mathbf{0}] & \ldots & [\mathbf{0}] \\ [\mathbf{0}] & \mathbf{a}_{i,2}^{(1)}, \mathbf{a}_{i,2}^{(2)}, \ldots, \mathbf{a}_{i,2}^{(M_2)}, & \ldots & [\mathbf{0}] \\ \vdots & \vdots & \ddots & \vdots \\ [\mathbf{0}] & [\mathbf{0}] & \ldots & \mathbf{a}_{i,K}^{(1)}, \mathbf{a}_{i,K}^{(2)}, \ldots, \mathbf{a}_{i,K}^{(M_K)} \end{bmatrix}. \tag{1}$$

(The rotation matrix will have dimensionality $n \times \sum_j M_j$.) To calculate the training set for classifier D_i we first rearrange the columns of R_i (the features) so that they correspond to the original features. Denote the rearranged rotation matrix R_i^a (size $N \times n$). Then, the training set for classifier D_i is XR_i^a.

Fig. 1 shows the pseudocode for the algorithm. We chose decision trees as the base classifiers because they are sensitive to rotation of the feature axes and still can be very accurate.

The feature extraction is based on Principal Component Analysis (PCA) [15]. It is well documented in the literature since the 1970s that PCA (known in pattern recognition as Karhunen-Loéve transformation) is not particularly suitable for feature extraction in classification because it does not include discriminatory information in

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

Training Phase

Given

- X: the objects in the training data set (an $N \times n$ matrix)
- Y: the labels of the training set (an $N \times 1$ matrix)
- L: the number of classifiers in the ensemble
- K: the number of subsets
- $\{\omega_1, \ldots, \omega_c\}$: the set of class labels

For $i = 1 \ldots L$

- Prepare the rotation matrix R_i^a:
  - Split $\mathbf{F}$ (the feature set) into K subsets: $\mathbf{F}_{i,j}$ (for $j = 1 \ldots K$)
  - For $j = 1 \ldots K$
    - Let $X_{i,j}$ be the data set X for the features in $\mathbf{F}_{i,j}$
    - Eliminate from $X_{i,j}$ a random subset of classes
    - Select a bootstrap sample from $X_{i,j}$ of size 75% of the number of objects in $X_{i,j}$. Denote the new set by $X'_{i,j}$
    - Apply PCA on $X'_{i,j}$ to obtain the coefficients in a matrix $C_{i,j}$
  - Arrange the $C_{i,j}$, for $j = 1 \ldots K$ in a rotation matrix R_i as in equation (1)
  - Construct R_i^a by rearranging the the columns of R_i so as to match the order of features in $\mathbf{F}$.
- Build classifier D_i using (XR_i^a, Y) as the training set

Classification Phase

- For a given $\mathbf{x}$, let $d_{i,j}(\mathbf{x}R_i^a)$ be the probability assigned by the classifier D_i to the hypothesis that $\mathbf{x}$ comes from class ω_j. Calculate the confidence for each class, ω_j, by the average combination method:

$$\mu_j(\mathbf{x}) = \frac{1}{L}\sum_{i=1}^{L} d_{i,j}(\mathbf{x}R_i^a), \quad j = 1, \ldots, c.$$

- Assign $\mathbf{x}$ to the class with the largest confidence.

Fig. 1. Pseudocode of the Rotation Forest ensemble method.

calculating the optimal rotation of the axes. Many alternative linear transformations have been suggested based on discrimination criteria [11], [14], [19], [43], [44]. Sometimes a simple random choice of the transformation matrix leads to classification accuracy superior to that when PCA is used. Fern and Brodley [9] advocate random projections rather than PCA for cluster ensembles. On the other hand, Skurichina and Duin [41] find that an ensemble based on PCA performs better than ensembles based on random feature selection. The ensemble proposed in their study is built using the principal components calculated on the whole data set. The first classifier uses the first M components, the next classifier uses the next M components, etc.

Tumer and Oza [42] also use PCA as a tool for ensemble generation, but in their study it is applied to reduce dimensionality. They propose an ensemble consisting of $L = c$ classifiers, where c is the number of classes. To produce diverse classifiers, they use different sets of extracted features. To train classifier D_i, they first single out class ω_i and run PCA on this data only. (The number of components to keep is a parameter of the algorithm.) The obtained transformation is applied on the whole data set and D_i is trained on these new extracted features to discriminate among all the original classes. This limits the ensemble size L to the number of classes.

We used PCA to determine its feasibility and find out whether it does contribute to increased accuracy and diversity. The examples in the literature exposing PCA as an inadequate method for feature extraction in classification problems are related to dimensionality reduction. As only few components are retained, there is a chance that the most relevant discriminatory components correspond to small variance and will be discarded. Here, we keep all the components, so the discriminatory information will be preserved even if it lies with the component responsible for the least variance. Keeping all the components does not mean that the classification will be easier in the new space of extracted features. However, even if the rotation does not contribute much to finding good discriminatory directions, it is valuable here as a diversifying heuristic. Pilot experiments with our Rotation Forest ensemble and random projections using all n features showed that although the results were competitive, they were not as good as the results with PCA.

The intended diversity in this model will come from the difference in the possible feature subsets. Using only the rotation heuristic, there are in total $T = \frac{n!}{K!(M!)^K}$ different partitions of the feature set into K subsets of size M, each giving raise to a classifier (recall that $n = KM$). If the ensemble consists of L classifiers, assuming that each partition of the feature set is equally probable, the probability that all classifiers will be different is $P(\text{different classifiers}) = \frac{T!}{(T-L)!T^L}$. For example, the chance to have all different classifiers in an ensemble of $L = 50$ classifiers for $K = 3$ and $n = 9$ is less than 0.01. Therefore, there is a need for an extra randomization of the ensemble. This can be done by applying PCA to a bootstrap sample from X, a random subset of X or a random selection of classes. In this study, we applied both these heuristics. A nonempty random subset of classes was chosen for each feature subset, and then a bootstrap sample of objects was drawn. PCA was applied on this new data set using only the features in

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

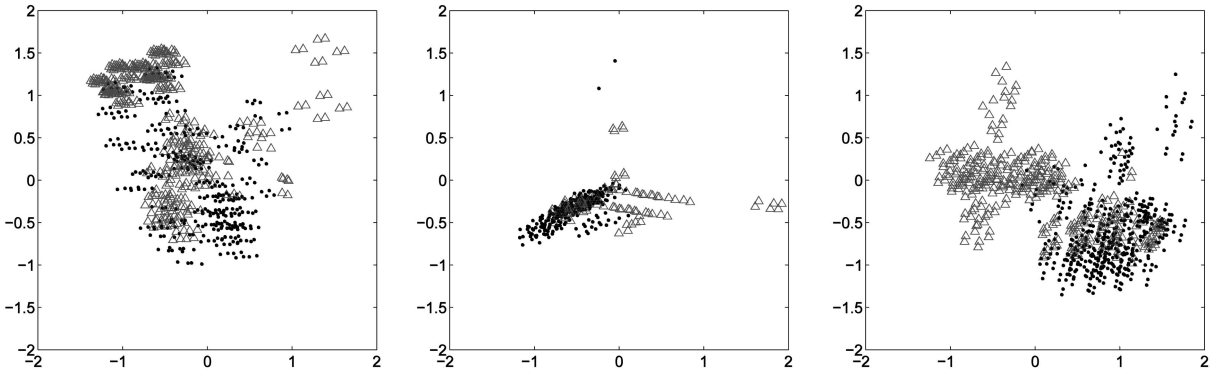


Fig. 2. Scatterplot of the "mushroom" data (22 nominal features). The binary representation of the features (121 in total) was randomly split into three subsets and PCA was applied on each subset. The scatterplots show the two classes in the space defined by the first two principal components.

the subset. Note, however, that after obtaining the projection matrix from all feature subsets, the *whole* data set was transformed and subsequently used for training a base classifier.

3 Experimental Validation

An experiment was set up to compare Rotation Forests with Bagging, AdaBoost, and Random Forests. In all ensemble methods, decision trees were used as the base classifier. The decision tree construction method was J48 from the WEKA library [47], a reimplementation of C4.5 [32], except for the Random Forest method. Random Forest constructs the tree in a different way so as to allow for a random choice of a feature at each node. The implementations of Bagging, AdaBoost.M1 (the version proposed in [12] for multiple classes) and Random Forests are also from that library. As PCA is defined for numeric features, discrete features were converted to numeric ones for Rotation Forests. Each categorical feature was replaced by s binary features encoded numerically as 0 and 1, where s is the number of possible categories of the feature.[3]

This encoding unifies all the inputs so that PCA can be carried out on any subsets of features. Kolenikov and Angeles [20] advise against using dummy variables to replace ordinal variables and subsequently apply PCA on these dummy variables. Their main argument is that the evaluation of the explained variance by the principal components in this case is largely imprecise. Their study is primarily looking at data analysis for devising scoring indices in economics. In our case, however, the estimate of the explained variance is irrelevant, as we keep all the principal components rather than discard the ones with smaller contribution to the variance. On the other hand, while the explained variance and the coefficients for the first few principal components are very important for a scoring index, they act merely as a rotation heuristic in our ensemble model. Finally, in the presence of mixed continuous-valued and categorical variables of nominal type (no ordering of the categories), binary encoding is the most natural way to unify the data for further processing. As neither interpretation of the extracted features, nor the total explained variance is a concern in Rotation Forest, we apply standard PCA. Each category is encoded by one bit. Clearly, this code contains redundancy as the sufficient number of bits would be $\log_2(\text{number of categories})$. Also, the binary features in the one-bit-per-category representation are dependent because only one of them can be 1 at a time. There are no strong arguments either way, so we adopted this encoding—it is easy, intuitive, and is the standard one implemented in WEKA [47]. An example of how PCA works on nominal variables is shown in Fig. 2. PCA has been applied to the "mushroom" data from the UCI repository [4], containing 22 nominal variables encoded as explained above into a total of 121 binary variables. The new feature set was split randomly into three subsets. Thus, some of the original variables were "split" themselves, so that the bits corresponding to the categories fell into different feature subsets. On each of the three feature subsets, a PCA was run. The three scatterplots show the data in the space of the first two principal components. All plots reveal a cluster structure indicating that a decision tree classifier would be a suitable choice in each of the new spaces.

Table 1 shows the characteristics of the 33 data sets used in this study. All of them are from the UCI repository [4]. We consider two versions of the "vowel" data, "vowel-c," and "vowel-n." In "vowel-c," the sex of the speaker and a speaker identifier were used and in "vowel-n" they were not. Because results with both versions have been reported in the literature, we used both versions too.

The experimental settings were as follows: The parameters of Bagging, AdaBoost, and Random Forests were kept at their default values in WEKA. For Random Forest, the number of features fo select from at each node is set at $\log_2(n) + 1$. For Rotation Forest, we did not fix K but fixed the number of features in each subset to be $M = 3$. If n did not divide by 3, the "remainder" subset was completed with 1 or 2 features, as necessary, randomly selected from the rest of the feature set.

The decision tree classifier, J48, implemented in WEKA uses an error-based pruning algorithm. The user can choose a confidence value to be used when pruning the tree. The default of 25 percent was found to work reasonably well, in most cases, so we left it unchanged in our experiments. This standard implementation was not suitable for Random Forests, so Rotation Forest was compared with Bagging and AdaBoost only.

The ensemble size L can be regarded as a hyper parameter of the ensemble method. It can be tuned through cross-validation or by using a separate validation set. L can

3. In the current implementation, this conversion is done as a preprocessing step, before the algorithm of Fig. 1. Hence, when the feature set is splitted in subsets, the binary features from a categorical feature can be assigned to different subsets. It could also be possible to do the conversion after the splitting.

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Characteristics of the 33 Data Sets Used in This Study

Data set	Classes	Objects	Discrete features	Continuous features
anneal	6	898	32	6
audiology	24	226	69	0
autos	7	205	10	16
balance-scale	3	625	0	4
breast-cancer	2	286	10	0
cleveland-14-heart	5	307	7	6
credit-rating	2	690	9	6
german-credit	2	1000	13	7
glass	7	214	0	9
heart-statlog	2	270	0	13
hepatitis	2	155	13	6
horse-colic	2	368	16	7
hungarian-14-heart	5	294	7	6
hypothyroid	4	3772	22	7
ionosphere	2	351	0	34
iris	3	150	0	4
labor	2	57	8	8
letter	26	20000	0	16
lymphography	4	148	15	3
pendigits	10	10992	0	16
pima-diabetes	2	768	0	8
primary-tumor	22	239	17	0
segment	7	2310	0	19
sonar	2	208	0	60
soybean	19	683	35	0
splice	3	3190	60	0
vehicle	4	846	0	18
vote	2	435	16	0
vowel-c	11	990	2	10
vowel-n	11	990	0	10
waveform	3	5000	0	40
wisconsin-breast	2	699	0	9
zoo	7	101	16	2

also be thought of as an indicator of the operating complexity of the ensemble. Then, we can choose the most accurate ensemble of a fixed complexity. As we are interested in ensembles of a small (fixed) size, we decided to train all ensemble methods with the same L. Hence, the results of the experiments will apply with this restriction in mind. If we let L be tuned, we could get a different ranking of the ensemble methods with respect to their accuracy as well as operational complexity.

Fig. 3 shows a percentage graph for Rotation Forest, Random Forest, AdaBoost and bagging for ensembles of unpruned decision trees, using one 10-fold cross-validation. The x-axis is the ensemble size, L, and the y-axis shows the percent of the data sets in which the respective method has been the one with the lowest error among the four methods. For example, we took all 33 data sets and looked at the accuracies of the four ensemble methods for ensemble size $L = 10$ (the results shown in Table 3 and discussed later). For each data set, one of the methods was declared "the winner." For example, for the "anneal" data set, Boosting shows the highest accuracy of 99.54 percent, so this method is the winner. The winnings for the methods were tallied across the 33 data sets. In this example, Rotation Forest scores 23 wins (69.70 percent), AdaBoost 8 wins (24.24 percent), and Bagging and Random Forest score 1 win each (3.03 percent). If we draw a vertical line in the figure at $L = 10$ and look at the

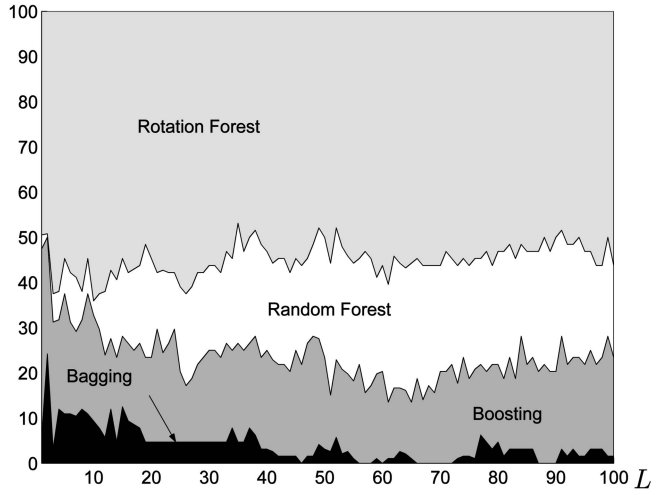


Fig. 3. Percentage diagram for the four studied ensemble methods with unpruned J48 trees.

segment from 0 to 100 percent, the bottom 3.03 percent will correspond to Bagging, the part from 3.03 to 27.27 will correspond to Boosting, the part from 27.27 to 30.30 will correspond to Random Forest, and the part from 30.30 to 100 percent will correspond to Rotation Forest. An integral-type measure of performance of a method would be the area corresponding to the percentage winnings for this method. As the figure shows, Rotation Forest is the most accurate method for all ensemble sizes, slightly better for smaller ensembles than for larger ensembles.

As explained before, we are interested in relatively small ensembles and so we fixed the ensemble size arbitrarily at $L = 10$. For each data set and ensemble method, 15 10-fold cross validations were performed. The average accuracies and the standard deviations are shown in Table 2. For reference, we display the accuracy of a single J48 tree as well. The results for which a significant difference with Rotation Forest was found are marked with a bullet or an open circle next to them. A bullet next to a result indicates that Rotation Forest was significantly *better* than the respective method (column) for the respective data set (row). An open circle next to a result indicates that Rotation Forest was significantly *worse* than the respective method.

For this comparison, we used a corrected estimate of the standard deviation proposed by Nadeau and Bengio [29] as implemented in WEKA, with a significance level of 5 percent. This estimate was developed as an answer to the criticism that the currently used statistical tests are unnecessarily "liberal," i.e., difference between a classifier's accuracy and a reference value might be found even though there is not any. The new estimate is more conservative and leads to a test with a specified size (chosen level of significance) and good power (low error rate in accepting that there is no difference if there is one). Instead of taking $\sigma_{\hat{\mu}} = \frac{\sigma_\mu}{\sqrt{T}}$ (T is the number of experiments), the authors propose

$$\sigma_{\hat{\mu}} = \sigma_\mu \sqrt{\frac{1}{T} + \frac{N_{\text{testing}}}{N_{\text{training}}}}, \tag{2}$$

where N_{training} and N_{testing} are the sizes of the training and the testing sets, respectively. Note that the comparison was

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

TABLE 2
Classification Accuracy and Standard Deviation of J48 and Ensemble Methods with Pruning

Data Set	Rotations J48	J48	Bagging J48	Boosting J48
anneal	98.93±0.95	98.61±1.06	98.89±0.92	99.58±0.71
audiology	79.80±6.92	77.24±7.04	81.03±7.36	84.90±7.07 ○
autos	82.50±8.66	82.34±9.22	82.69±8.60	85.31±6.99
balance-scale	90.33±2.52	77.82±3.69 ●	81.85±3.74 ●	78.46±4.07 ●
breast-cancer	72.66±6.71	74.19±6.05	72.65±6.12	66.88±7.37 ●
cleveland-14-heart	82.85±6.26	76.71±6.84 ●	79.21±6.74	79.38±6.99
credit-rating	86.13±3.88	85.63±4.12	85.78±4.02	83.86±4.35
german-credit	74.10±3.93	71.09±3.53 ●	73.75±3.62	71.01±3.93 ●
glass	74.27±8.11	67.55±9.33 ●	73.97±9.41	75.20±8.26
heart-statlog	82.25±6.43	78.22±7.20	80.74±6.66	78.27±7.20
hepatitis	82.80±8.91	79.58±9.28	81.24±8.22	82.46±8.00
horse-colic	84.73±5.44	85.16±5.70	85.41±5.70	81.63±6.11
hungarian-14-heart	80.28±6.33	80.08±7.65	79.62±6.70	78.75±6.65
hypothyroid	99.56±0.35	99.53±0.35	99.58±0.32	99.64±0.30
ionosphere	93.88±3.68	89.91±4.57 ●	92.25±3.80	93.18±4.02
iris	95.73±5.20	94.89±5.03	94.67±5.12	94.27±5.18
labor	91.56±11.91	79.56±15.78●	83.13±15.20	87.31±13.36
letter	95.48±0.47	88.04±0.73 ●	92.72±0.63 ●	95.53±0.47
lymphography	83.99±8.33	76.37±11.09●	77.97±10.22●	81.73±8.61
pendigits	99.20±0.26	96.46±0.56 ●	97.93±0.47 ●	99.02±0.30
pima-diabetes	76.48±4.44	74.38±4.91	75.65±4.45	71.96±4.53 ●
primary-tumor	45.06±6.40	41.71±6.83	43.74±6.76	41.87±6.53
segment	98.05±0.95	96.79±1.28 ●	97.49±1.07	98.14±0.89
sonar	83.56±7.84	73.98±8.67 ●	78.31±9.11	79.79±8.63
soybean	94.77±2.36	91.90±3.11 ●	92.73±2.87 ●	92.74±2.82 ●
splice	95.47±1.15	94.17±1.22 ●	94.43±1.26 ●	94.60±1.15 ●
vehicle	78.05±3.64	72.33±4.42 ●	74.45±4.18 ●	75.78±4.19
vote	96.26±2.79	96.49±2.65	96.37±2.54	95.34±3.11
vowel-c	96.89±1.74	79.62±4.17 ●	90.20±3.16 ●	92.77±2.77 ●
vowel-n	95.68±1.95	79.16±4.58 ●	89.45±3.22 ●	92.13±2.84 ●
waveform	83.93±1.69	75.27±2.00 ●	81.75±1.70 ●	81.34±1.88 ●
wisconsin-breast-cancer	97.04±1.94	94.87±2.69 ●	95.99±2.44	96.06±2.27
zoo	92.15±8.22	92.56±7.04	93.30±7.07	96.38±5.75
(Win/Tie/Loss)		(0/16/18)	(0/24/10)	(1/24/9)

○ Rotation Forest is significantly worse, ● Rotation Forest is significantly better, level of significance 0.05

carried out using all the $T = 150$ testing accuracies per method and data set (15×10-fold CV), not the 15 accuracies of each CV. The standard deviations reported in Table 2, σ_μ, are calculated across the 150 testing accuracies.

It has been found that pruning is not always beneficial for the ensemble [3], [8]. Hence, the experiments were repeated with pruning being suspended. The results are shown in Table 3. Here, it was fair to include Random Forests, because it does not use pruning. The results from a statistical comparison are again indicated by bullets and open circles.

Fig. 4 gives a graphical overview of the results in Table 2 (subplot a) and Table 3 (subplot b). On the y-axis is the accuracy of the Rotation Forest ensemble while on the x-axis is the best accuracy among the competing ensemble methods.[4] If Rotation Forest was better than the other ensembles for each data set, all the points on the graph would lie above the dashed diagonal line which marks the equivalent scores. The data sets for which the differences were particularly large are labeled in the graphs. As most of the points in both subplots lie above the diagonal line, the figure demonstrates the advantage of Rotation Forest.

Table 4 shows a summary of the comparisons among the methods. The entry $a_{i,j}$ displays the number of times when the method of the column (j) has a better result than the method of the row (i). The number in the parentheses shows in how many of these differences have been statistically significant. For example, Rotation Forest with pruned trees has been better than AdaBoost with pruned trees in 25 of the 33 comparisons and worse in 8. The numbers in the parentheses show that, in nine cases, the difference in favor of Rotation Forest has been statistically significant; hence, the value 25(9) in row 3, column 4 of the table. In one of the opposite eight cases, AdaBoost was significantly better than Rotation Forest (the audiology data set), i.e., the entry in row 4, column 3 is 8(1).[5]

Table 5 shows a ranking of the methods according to the difference between the number of times each method has been significantly better and significantly worse than another method. Here, we use all pairwise comparisons as summarized in Table 4. For example, the sum of the numbers in the brackets in the *column* corresponding to Rotation Forest using pruned trees in Table 4 is 84. The sum of the numbers in the brackets in the *row* corresponding to

4. The result for data set "primary tumor" is left out of the plots because the accuracy with all ensemble methods is very low, the corresponding point represents an outlier.

5. The sum "better + worse" for each pair of methods should be 33 as there are 33 data sets (25 + 8 for Rotation Forest and AdaBoost with pruned trees). The only exception is Rotation Forest with pruning versus Rotation Forest without pruning. For this case, "better + worse" counts are $15 + 17 = 32$. This happens because the accuracies were exactly the same for the iris data.

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

TABLE 3
Classification Accuracy and Standard Deviation of J48 and Ensemble Methods *without* Pruning

Data Set	Rotations J48	J48	Bagging J48	Boosting J48	Random Forest
anneal	99.01±0.93	98.62±1.01	98.98±0.93	99.54±0.68	99.38±0.78
audiology	79.83±6.93	76.33±7.45 ●	81.12±7.35	83.30±6.99	76.58±7.94
autos	82.56±8.66	82.86±9.25	84.12±8.42	84.61±7.93	81.95±7.85
balance-scale	90.26±2.62	79.43±4.01 ●	81.39±3.70 ●	76.82±4.14 ●	80.28±3.80 ●
breast-cancer	72.07±6.54	68.00±7.43	69.48±7.17	66.12±7.81 ●	69.00±7.31
cleveland-14-heart	82.61±6.12	76.49±6.91 ●	79.70±6.01	79.20±7.25	80.34±6.47
credit-rating	86.00±3.90	82.50±4.24 ●	85.17±4.34	84.02±3.98	85.15±4.23
glass	74.33±8.06	67.77±9.70 ●	73.85±9.34	76.23±9.09	75.65±8.42
german-credit	73.87±3.89	67.89±3.95 ●	72.08±3.63	71.95±4.32	73.57±3.38
heart-statlog	82.37±6.45	76.69±7.51 ●	80.44±6.84	79.38±7.40	80.86±6.53
hepatitis	82.92±8.88	78.95±9.27	80.68±8.89	82.45±8.17	83.04±8.07
horse-colic	84.80±5.35	82.16±5.89	84.80±5.96	81.05±6.20	84.96±5.43
hungarian-14-heart	79.57±6.45	78.85±7.30	78.74±6.65	79.08±7.00	79.28±6.31
hypothyroid	99.57±0.33	99.51±0.37	99.59±0.30	99.65±0.30	99.18±0.46 ●
ionosphere	93.88±3.76	89.97±4.55 ●	92.29±3.79	93.01±3.97	92.84±3.89
iris	95.73±5.20	94.93±4.99	94.58±5.15	94.36±5.22	94.13±5.18
labor	91.69±11.89	79.84±14.57●	84.31±14.44	87.20±13.81	87.00±13.45
letter	95.54±0.47	88.02±0.75 ●	92.85±0.65 ●	95.44±0.50	94.52±0.49 ●
lymphography	84.27±8.35	75.64±11.12●	78.97±10.32	82.40±9.73	81.28±8.58
pendigits	99.21±0.25	96.46±0.57 ●	97.99±0.44 ●	99.01±0.28 ●	98.81±0.29 ●
pima-diabetes	76.39±4.43	73.85±4.94	75.59±4.54	72.49±5.08 ●	74.78±4.42
primary-tumor	44.37±6.56	42.42±7.57	42.79±6.92	41.64±6.94	41.56±6.50
segment	98.05±0.95	96.81±1.26 ●	97.58±1.05	98.25±0.80	97.71±1.06
sonar	83.49±7.88	73.82±8.71 ●	78.34±9.14	79.95±9.51	80.75±7.84
soybean	94.17±2.47	90.67±3.34 ●	91.88±3.15 ●	92.44±2.76	91.92±2.83 ●
splice	95.49±1.13	92.20±1.37 ●	94.25±1.20 ●	94.11±1.23 ●	90.07±1.79 ●
vehicle	77.95±3.74	72.38±4.25 ●	74.70±4.07 ●	76.44±4.01	74.37±4.43 ●
vote	96.08±2.88	95.71±2.93	96.43±2.47	95.22±3.19	95.74±2.75
vowel-c	96.87±1.76	81.26±4.18 ●	91.72±2.89 ●	94.15±2.42 ●	95.59±2.23
vowel-n	95.77±1.94	79.22±4.59 ●	89.52±3.27 ●	91.93±2.72 ●	92.37±2.73 ●
waveform	83.94±1.72	75.14±1.99 ●	81.78±1.74 ●	81.45±1.71 ●	81.89±1.74 ●
wisconsin-breast-cancer	97.02±1.93	94.30±2.74 ●	95.82±2.54 ●	95.97±2.11	95.75±2.14 ●
zoo	92.35±8.04	93.42±6.93	93.50±7.11	97.04±5.21 ○	95.83±6.02
(Win/Tie/Loss)		(0/13/21)	(0/24/10)	(1/25/8)	(0/24/10)

○ Rotation Forest is significantly worse, ● Rotation Forest is significantly better, level of significance 0.05

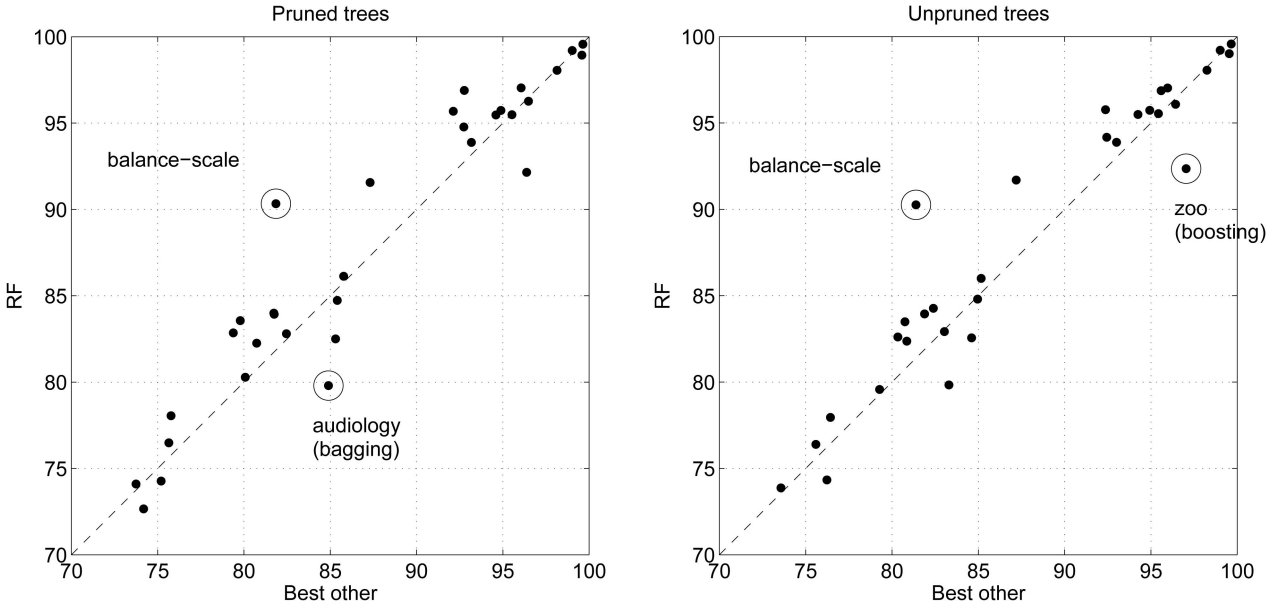


Fig. 4. Comparison of accuracy of Rotation Forest ensemble (RF) and the best accuracy from any of a single tree, Bagging, Boosting, and Random Forest ensembles.

Rotation Forest is 2. These are used in Table 5 to calculate the nondominance ranking of Rotation Forest $(84 - 2)$.

Tables 4 and 5 demonstrate the advantage of Rotation Forest compared with the best benchmark classifier ensemble methods: Bagging, AdaBoost, and Random Forest. The dominance of Rotation Forest was established by a large margin leaving AdaBoost, Bagging, and Random Forest in a group with much smaller differences among them.

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

TABLE 4
Summary of Results

	Pruned trees				Unpruned trees				
	J48	Bagging	AdaBoost	Rotation Forest	J48	Bagging	AdaBoost	Random Forest	Rotation Forest
Pruned trees									
J48	-	29 (9)	25 (12)	29 (18)	14 (2)	26 (9)	23 (12)	22 (8)	28 (18)
Bagging	4 (0)	-	21 (7)	27 (10)	3 (0)	18 (2)	17 (6)	16 (4)	25 (9)
AdaBoost	8 (1)	12 (3)	-	25 (9)	8 (0)	12 (2)	15 (0)	16 (1)	26 (7)
Rotation Forest	4 (0)	6 (0)	8 (1)	-	2 (0)	7 (0)	7 (1)	5 (0)	17 (0)
Unpruned trees									
J48	19 (5)	30 (14)	25 (14)	31 (19)	-	31 (12)	26 (13)	28 (9)	31 (21)
Bagging	7 (1)	15 (0)	21 (4)	26 (10)	2 (0)	-	20 (5)	20 (4)	28 (10)
AdaBoost	10 (1)	16 (3)	18 (0)	26 (8)	7 (1)	13 (1)	-	15 (1)	26 (8)
Random Forest	11 (2)	17 (2)	17 (5)	28 (10)	5 (2)	13 (2)	18 (4)	-	28 (10)
Rotation Forest	5 (0)	8 (0)	7 (1)	15 (0)	2 (0)	5 (0)	7 (1)	5 (0)	-

The entry $a_{i,j}$ shows the number of times method of the column (j) has a better result than the method of the row (i). The number in the parentheses shows in how many of these differences have been statistically significant.

To analyze the reasons for the success of Rotation Forests, we look at the diversity-accuracy pattern of the generated ensembles.

4 Diversity-Error Diagrams

Margineantu and Dietterich suggested a visualization means for classifier ensembles [26]. Pairwise diversity measures were chosen because "diversity" is intuitively clear for two variables (two classifier outputs in this case). Dependence between two variables can be measured as "deviation from independence," using a correlation coefficient or an appropriate statistic for nominal variables (class labels). For more than two variables, such a measure of diversity is not easy to define [21], [22]. The pair-wise diversity measure used in [26] is the interrater agreement, kappa (κ). Kappa evaluates the level of agreement between two classifier outputs while correcting for chance [10]. For c class labels, κ is defined on the $c \times c$ coincidence matrix $\mathcal{M}$ of the two classifiers. The entry $m_{k,s}$ of $\mathcal{M}$ is the proportion of the data set used for testing, which D_i labels as ω_k and D_j labels as ω_s. The agreement between D_i and D_j is given by

$$\kappa_{i,j} = \frac{\sum_k m_{kk} - \text{ABC}}{1 - \text{ABC}}, \tag{3}$$

where $\sum_k m_{kk}$ is the observed agreement between the classifiers and 'ABC' is "agreement-by-chance"

$$\text{ABC} = \sum_k \left(\sum_s m_{k,s} \right) \left(\sum_s m_{s,k} \right). \tag{4}$$

Low values of κ signify high disagreement and, hence, high diversity. If calculated for two classes,

$$\kappa_{i,j} = \frac{2(m_{1,1}m_{2,2} - m_{1,2}m_{2,1})}{(m_{1,1} + m_{1,2})(m_{1,1} + m_{2,1}) + (m_{1,2} + m_{2,2})(m_{2,1} + m_{2,2})}. \tag{5}$$

If the classifiers produce identical class labels, only the main diagonal of $\mathcal{M}$ will contain nonzero elements and $\kappa = (1 - \text{ABC})/(1 - \text{ABC}) = 1$. If the classifiers are independent, their agreement will be the same as the agreement by chance (ABC) and $\kappa = 0$. Independence is not necessarily the best scenario in multiple classifier systems [24]. Even more desirable is "negative dependence," $\kappa < 0$, whereby classifiers commit related errors so that when one classifier is wrong, the other has more than random chance of being correct.

An ensemble of L classifier generates $L(L-1)/2$ pairs of classifiers D_i, D_j. On the x-axis of a kappa-error diagram is the κ for the pair and on the y-axis is the averaged individual error of D_i and D_j, $E_{i,j} = \frac{E_i + E_j}{2}$. As small values of κ indicate better diversity and small values of $E_{i,j}$ indicate better accuracy, the most desirable pairs of classifiers will lie in the bottom left corner.

Fig. 5 shows the kappa-error diagrams for five data sets. All ensembles consisted of 100 classifiers (decision trees),

TABLE 5
Ranking of the Methods Using the Significant Differences from All Pairwise Comparisons

Method	Dominance rank (Wins−Losses)	Wins	Losses
Rotations J48 pruned	82	84	2
Rotations J48 unpruned	81	83	2
AdaBoostM1 J48 pruned	21	44	23
AdaBoostM1 J48 unpruned	19	42	23
Bagging J48 unpruned	−6	28	34
Bagging J48 pruned	−7	31	38
Random Forest	−10	27	37
J48 pruned	−78	10	88
J48 unpruned	−102	5	107

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

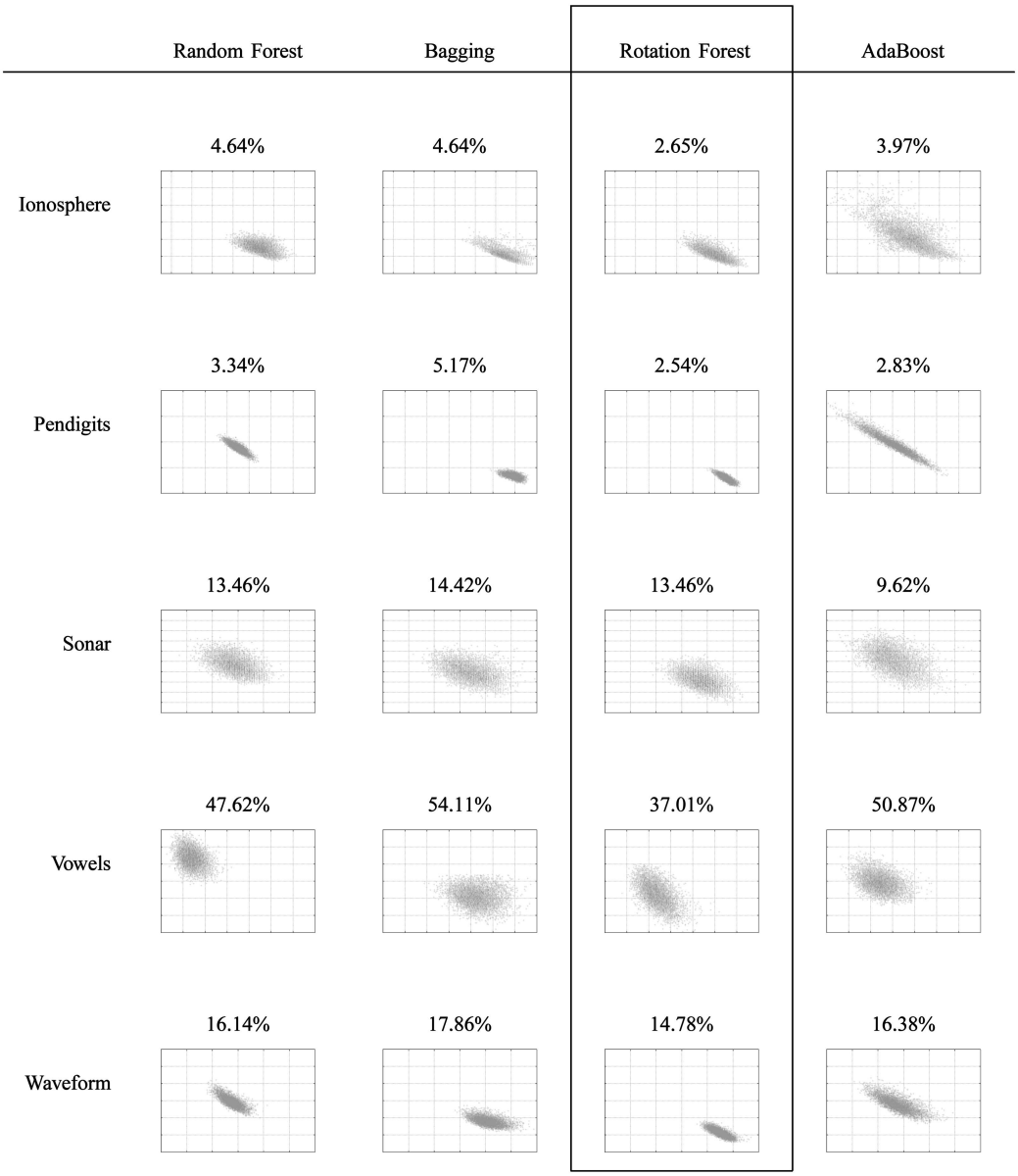


Fig. 5. κ-Error Diagrams. x-axis = κ, y-axis = $E_{i,j}$ (average error of the pair of classifiers). Axes scales are constant for each row. The *ensemble error* on the testing set is displayed above the plot.

therefore there are 4,950 dots in each plot. We chose these sets because they had a predefined partition of the objects into training and testing subsets. The ensembles were trained on the training sets and the kappa-error diagrams were calculated on the testing sets. All ensembles consisted of unpruned trees; the diagrams for pruned ensembles were similar. The scales for κ and $E_{i,j}$ are the same for each given data set but may vary from one data set to another. It was important to show the relative position of the clouds of points for each data set.

Fig. 6 plots the centroids of the clouds of kappa-error points in the same plot for each data set. This enables a visual evaluation of the relative positions of the clouds for the respective ensemble methods. The axes of the plots are rescaled so that the relative positions of the centroids are clearly visible. Note that with some data sets the clouds are heavily overlapping and the distances between the centroids are small. Since the rescaling magnifies the distances, Figs. 5 and 6 cannot be easily compared.

Shown above each plot in Fig. 5 is the *ensemble error* on the testing set. To help read the graphs, take row 1 as an example. AdaBoost spans the largest diversity range but many classifier pairs have large errors. The other three ensemble methods have similar accuracies, with Random Forest being less accurate than both Bagging and Rotation Forest (elevated on the y-axis). Random Forest has better diversity than Bagging. Rotation Forest has accuracy approximately equal to that of Bagging, but better diversity (position of the x-axis; the cloud of points for Rotation Forest is situated to the left of the cloud for bagging). Thus, Rotation Forest takes "the best of both worlds." However, the improvement on the diversity-accuracy pattern was not sufficient to fetch statistically significant difference in the ensemble performances in the general experiment. (Note that the picture is based on one ensemble only.)

On the other hand, statistically significant difference was observed with the Pendigits data set shown as row 2 in Fig. 5 and in second subplot in Fig. 6. The Rotation Forest ensemble

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

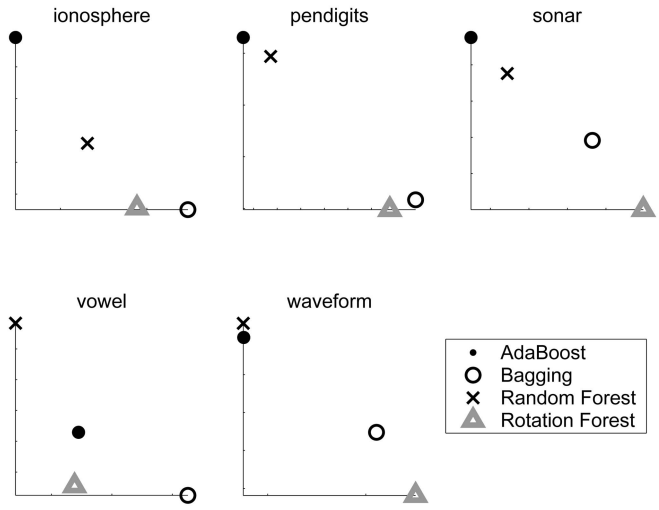


Fig. 6. Centroids of the kappa-error clouds for the five data.

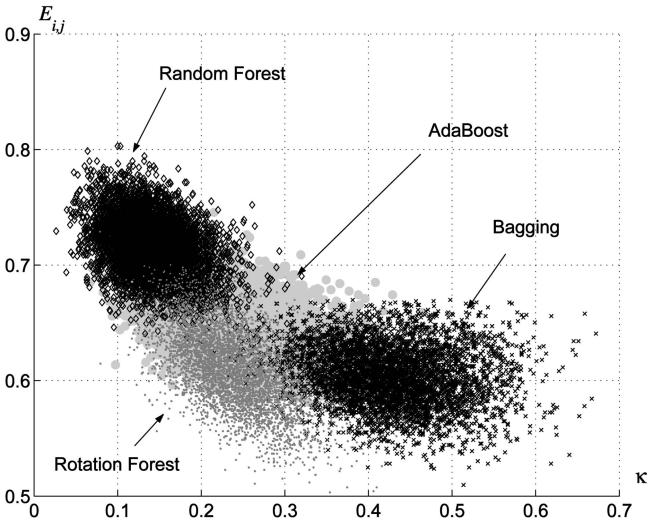


Fig. 7. Kappa-error diagrams for the vowel-n data set.

is less diverse than AdaBoost and Random Forest but rather more accurate. It is slightly more accurate and slightly more diverse than Bagging (seen in Fig. 6). It seems that accuracy pays off with this data set as the differences in ensemble performances are found to be statistically significant in favor of Rotation Forest.[6]

Fig. 7 shows the kappa-error diagrams for the four ensemble methods on the same plot for the vowel-n data set. This is one of the unusual disposition of the clouds of points because Rotation Forest is substantially more diverse than Bagging. The plot indicates that Rotation Forest has the potential to improve on diversity significantly without compromising the individual accuracy. The testing results shown in Fig. 5 reveal that Rotation Forest outperforms the other methods by a large margin.

A more typical example is shown in Fig. 8. The graph shows that Rotation Forest is not as diverse as the other ensembles but clearly has the most accurate classifiers in it. This comes to show that individual accuracy is probably the more crucial component of the tandem diversity-accuracy, contrary to the diversifying strategies which underpin

6. Note that "accuracy" in this context means the accuracy of the individual classifiers in the ensemble.

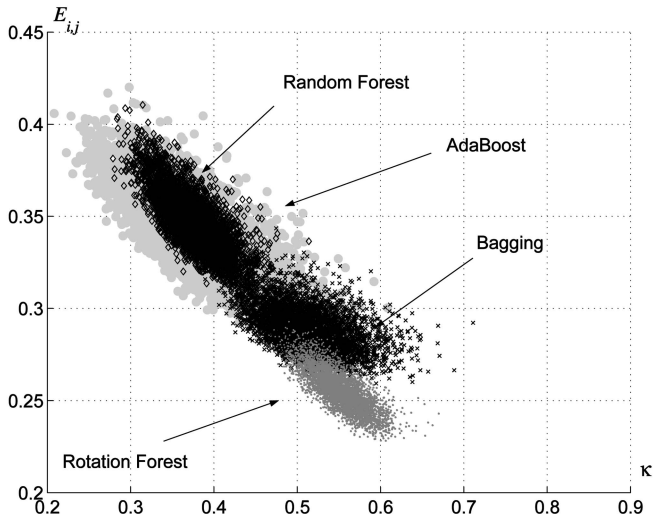


Fig. 8. Kappa-error diagrams for the waveform data set.

AdaBoost and Random Forests, and which have proven to be extremely successful. It seems that there is a scope for ensemble methods "on the other side of Bagging," i.e., ensembles based on reasonably diverse but markedly accurate members.

In general, Rotation Forest is similar to Bagging. Like Bagging, Rotation Forest is more accurate and less diverse than both AdaBoost and Random Forest. So, it seems that Rotation Forest has a place in the ensemble landscape next to Bagging, slightly improving on both its diversity and accuracy. The results in the previous section show that this seemingly minimal improvement on the diversity-accuracy pattern materializes in significantly better ensembles. This result has, among others, the implication that efforts towards even small improvements on diversity-accuracy pattern may bring unexpected benefits.

Below, we list a few caveats and our comments on these.

- Rotation Forest has an extra parameter which controls the sizes of the feature subsets or equivalently the number of feature subsets. We fixed the size of feature subsets to $M = 3$ here. We did not tune the hyperparameters of any of the ensemble methods in the experiment, including Rotation Forest. We acknowledge that a thorough experimental comparison of a set of methods needs tuning each of the methods to its best for every data set. Our reason for using the standard implementations in WEKA is three-fold. First, our intention was to gain initial feel of the merit of the proposed method. If it failed against the standard implementations of other ensemble methods, then no further investigations would be worthwhile. Second, standard implementations are meant to be general enough to work reasonably well across a variety of problems. The lack of fine-tuning of the methods is compensated by the diversity of the chosen data sets. The data sets were chosen randomly, not intentionally favoring one method or another. Third, we would like our experiment to be easily reproducible by others.
- The data sets used in the experiments were all taken from the UCI Machine Learning Repository. They

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

represent a variety of problems but do not include very large-scale data sets with the number of features or objects in a scale of millions. Examples of such data sets are DNA microarray data, searches of the Web and image databases, time series data, data from financial transactions, etc. Modifications of the standard classification methods and algorithms are needed in order to account for the specifics of such large data sets. This applies to ensemble methods as well including the rotation forest method. Thus, the results reported here are valid for data sets of small to moderate sizes.

- Random Forest offers as a by-product a way to order the features by their importance. Rotation Forest does not share this capability.
- Here, we used the same ensemble size L for all methods. It is known that bagging fares better for large L. On the other hand, AdaBoost would benefit from tuning L. Our results apply to ensembles of fixed L ($L = 10$ for the statistical comparisons), which, admittedly, is quite small by the ensemble standards. For larger L, we expect the differences to fade away, as indicated by Fig. 3. It is not clear what the outcome would be if L was treated as hyperparameter and tuned for all ensemble methods compared here.

5 Conclusions and Future Work

A novel method for generating ensembles of classifiers has been proposed. It consists in splitting the feature set into K subsets, running principal component analysis (PCA) separately on each subset and then reassembling a new extracted feature set while keeping all the components. The data is transformed linearly into the new features. A decision tree classifier is trained with this data set. Different splits of the feature set will lead to different rotations. Thus diverse classifiers are obtained. On the other hand, the information about the scatter of the data is completely preserved in the new space of extracted features. In this way, accurate individual classifiers are built. Thus, we target diversity and accuracy together.

Since PCA is a simple rotation of the coordinate axes and the base classifier model is a decision tree we call the proposed ensemble method *Rotation Forest*. Decision trees were used here because this classifier model is sensitive to rotation of the axes, yet sufficiently accurate.

Rotation Forest was compared with the standard implementations of Bagging, AdaBoost, and Random Forest available in WEKA. The experimental results with 33 data sets from UCI Machine Learning Repository showed that Rotation Forest outperformed all three methods by a large margin. We looked for an explanation of this improvement by using kappa-error diagrams. It transpired that Rotation Forest has similar diversity-accuracy pattern to Bagging, but is slightly more accurate and slightly more diverse than it. This seemingly marginal improvement fetched statistically significant differences in favor of Rotation Forest. Future studies and developments of Rotation Forest include, but are not limited to:

1. Evaluation of the sensitivity of the algorithm to the choice of M (or alternatively the number of feature subsets K) and to the choice of the ensemble size, L.
2. Application of Rotation Forest together with other ensemble approaches.
3. Trying a different base classifier model, e.g., Naïve Bayes and neural networks, to see whether the "forest" component is a necessity for the ensemble success.
4. Examining the effect of randomly pruning classes and taking a bootstrap sample for each feature subset, prior to applying PCA. Removing these two heuristics might lead to more similar PCA projections being obtained. It is interesting to find out whether or not this will have an adverse effect on the performance of Rotation Forest.
5. A different feature extraction algorithm can be used in the place of the PCA.

While we view the proposed ensemble model as a general framework, variants thereof will have to be tailored to address the application specifics.

Acknowledgments

The authors would like to thank the reviewers for their comments. This work was partially supported by Spanish Ministry of Education and Culture, through grant DPI2005-08498, Junta Castilla y Leon VA088A05 and a Mobility Grant from the University of Burgos.

References

[1] E.L. Allwein, R.E. Schapire, and Y. Singer, "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers," *J. Machine Learning Research,* vol. 1, pp. 113-141, 2000.

[2] R.E. Banfield, L.O. Hall, K.W. Bowyer, D. Bhadoria, W.P. Kegelmeyer, and S. Eschrich, "A Comparison of Ensemble Creation Techniques," *Proc Fifth Int'l Workshop Multiple Classifier Systems (MCS '04),* 2004.

[3] E. Bauer and R. Kohavi, "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants," *Machine Learning,* vol. 36, nos. 1-2, pp. 105-139, 1999.

[4] C.L. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases," 1998, http://www.ics.uci.edu/mlearn/MLRepository.html.

[5] L. Breiman, "Bagging Predictors," *Machine Learning,* vol. 24, no. 2, pp. 123-140, 1996.

[6] L. Breiman, "Arcing Classifiers," *Annals of Statistics,* vol. 26, no. 3, pp. 801-849, 1998.

[7] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[8] T.G. Dietterich, "Ensemble Methods in Machine Learning," *Proc. Conf. Multiple Classifier Systems,* pp. 1-15, 2000.

[9] X.Z. Fern and C.E. Brodley, "Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach," *Proc. 20th Int'l Conf. Machine Learning (ICML),* pp. 186-193, 2003.

[10] J.L. Fleiss, *Statistical Methods for Rates and Proportions.* John Wiley and Sons, 1981.

[11] F.H. Foley and J.W. Sammon, "An Optimal Set of Discriminant Vectors," *IEEE Trans. Computers,* vol. 24, no. 3, pp. 281-289, Mar. 1975.

[12] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.

[13] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics,* vol. 28, no. 2, pp. 337-374, 2000.

[14] K. Fukunaga and W.L.G. Koontz, "Application of the Karhunen-Loeve Expansion to Feature Selection and Ordering," *IEEE Trans. Computers,* vol. 19, no. 4, pp. 311-318, Apr. 1970.

[15] J. Han and M. Kamber, *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2001.

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.

[16] L.K. Hansen and P. Salamon, "Neural Network Ensembles," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 10, pp. 993-1001, Oct. 1990.
[17] T.K. Ho, "The Random Subspace Method for Constructing Decision Forests," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 832-844, Aug. 1998.
[18] T.K. Ho, "A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors," *Pattern Analysis and Applications,* vol. 5, pp. 102-112, 2002.
[19] J. Kittler and P.C. Young, "A New Approach to Feature Selection Based on the Karhunen-Loeve Expansion," *Pattern Recognition,* vol. 5, no. 4, pp. 335-352, Dec. 1973.
[20] S. Kolenikov and G. Angeles, "The Use of Discrete Data in PCA: Theory, Simulations, and Applications to Socioeconomic Indices," *Proc. 2004 Joint Statistical Meeting,* 2004.
[21] L.I. Kuncheva, *Combining Pattern Classifiers. Methods and Algorithms.* John Wiley and Sons, 2004.
[22] L.I. Kuncheva and C.J. Whitaker, "Measures of Diversity in Classifier Ensembles," *Machine Learning,* vol. 51, pp. 181-207, 2003.
[23] L.I. Kuncheva, "Diversity in Multiple Classifier Systems (editorial)," *Information Fusion,* vol. 6, no. 1, pp. 3-4, 2004.
[24] L.I. Kuncheva, C.J. Whitaker, C.A. Shipp, and R.P.W. Duin, "Is Independence Good for Combining Classifiers?" *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 2, pp. 169-171, 2000.
[25] P.M. Long and V.B. Vega, "Boosting and Microarray Data," *Machine Learning,* vol. 52, pp. 31-44, 2003.
[26] D.D. Margineantu and T.G. Dietterich, "Pruning Adaptive Boosting," *Proc. 14th Int'l Conf. Machine Learning,* pp. 211-218, 1997.
[27] L. Mason, P.L. Bartlet, and J. Baxter, "Improved Generalization through Explicit Optimization of Margins," *Machine Learning,* vol. 38, no. 3, pp. 243-255, 2000.
[28] P. Melville, N. Shah, L. Mihalkova, and R.J. Mooney, "Experiments with Ensembles with Missing and Noisy Data," *Proc Fifth Int'l Workshop Multiple Classifier Systems,* pp. 293-302, 2004.
[29] C. Nadeau and Y. Bengio, "Inference for the Generalization Error," *Machine Learning,* vol. 62, pp. 239-281, 2003.
[30] N.C. Oza, "Boosting with Averaged Weight Vectors," *Proc Fourth Int'l Workshop Multiple Classifier Systems (MCS 2003),* 2003.
[31] *Multiple Classifier Systems, Proc. Sixth Int'l Workshop, MCS 2005,* N.C. Oza et al., eds., 2005.
[32] J.R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
[33] *Proc. First Int'l Workshop Multiple Classifier Systems (MCS 2000),* F. Roli and J. Kittler, eds., 2001.
[34] *Proc. Second Int'l Workshop Multiple Classifier Systems (MCS 2001),* F. Roli and J. Kittler, eds., 2001.
[35] *Proc. Third Int'l Workshop Multiple Classifier Systems (MCS 2002),* F. Roli and J. Kittler, eds., 2002.
[36] *Proc. Fifth Int'l Workshop Multiple Classifier Systems (MCS 2004),* F. Roli et al., eds., 2004.
[37] R.E. Schapire, "Theoretical Views of Boosting," *Proc. Fourth European Conf. Computational Learning Theory,* pp. 1-10, 1999.
[38] R.E. Schapire, "The Boosting Approach to Machine Learning: An Overview," *Proc. MSRI Workshop Nonlinear Estimation and Classification,* 2002.
[39] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *Annals of Statistics,* vol. 26, no. 5, pp. 1651-1686, 1998.
[40] R.E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-Rated Predictions," *Machine Learning,* vol. 37, no. 3, pp. 397-336, 1999.
[41] M. Skurichina and R.P.W. Duin, "Combining Feature Subsets in Feature Selection," *Proc. Sixth Int'l Workshop Multiple Classifier Systems, (MCS '05),* pp. 165-175, 2005.
[42] K. Tumer and N.C. Oza, "Input Decimated Ensembles," *Pattern Analysis Applications,* vol. 6, pp. 65-77, 2003.
[43] F. van der Heijden, R.P.W. Duin, D. de Ridder, and D.M.J. Tax, *Classification, Parameter Estimation and State Estimation.* Wiley, 2004.
[44] A. Webb, *Statistical Pattern Recognition.* London: Arnold, 1999.
[45] G.I. Webb, "MultiBoosting: A Technique for Combining Boosting and Wagging," *Machine Learning,* vol. 40, no. 2, pp. 159-196, 2000.
[46] *Proc. Fourth Int'l Workshop Multiple Classifier Systems (MCS 2003),* T. Windeatt and F. Roli, eds., 2003.
[47] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques,* second ed. Morgan Kaufmann, 2005.



Juan J. Rodríguez received the BS, MS, and PhD degrees in computer science from the University of Valladolid, Spain, in 1994, 1998, and 2004, respectively. He worked with the Department of Computer Science, University of Valladolid from 1995 to 2000. Currently, he is working with the Department of Civil Engineering, University of Burgos, Spain, where he is an associate professor. His main interests are machine learning, data mining, and pattern recognition. He is a member of the IEEE Computer Society.



Ludmila I. Kuncheva received the MSc degree from the Technical University, Sofia, in 1982 and the PhD degree from the Bulgarian Academy of Sciences in 1987. Until 1997, she worked at the Central Laboratory of Biomedical Engineering, Bulgarian Academy of Sciences, as a senior research associate. She is currently a reader at the School of Informatics, University of Wales, Bangor, United Kingdom. Her interests include pattern recognition, classifier combination, diversity measures, and nearest neighbor classifiers. She has published more than 100 research papers and two books. She won of the best paper award for 2006 in *IEEE Transactions on Fuzzy Systems* and the Sage best transaction paper award for 2003 across *IEEE Transactions on Systems, Man, and Cybernetics, A, B, and C*. She has served as an associate editor for the *IEEE Transactions on Fuzzy Systems* and is currently an associate editor for *IEEE Transactions on Pattern Analysis and Machine Intelligence*. She is a member of the IEEE.



Carlos J. Alonso received the BS degree in science and the PhD degree in physics, from the University of Valladolid, Valladolid, Spain, in 1985 and 1990, respectively. Currently, he is an associate professor in the Departamento de Informática, University of Valladolid. He has been working in different nationally funded projects related to supervision and diagnosis of continuous industrial environments. His main research interests are knowledge-based systems for supervision and diagnosis of dynamic systems, model-based diagnosis, knowledge engineering, and machine learning.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: UNIVERSIDAD DE CORDOBA. Downloaded on February 23,2024 at 10:55:31 UTC from IEEE Xplore. Restrictions apply.