Classifier Ensembles with a Random Linear Oracle

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

Abstract—We propose a combined fusion-selection approach to classifier ensemble design. Each classifier in the ensemble is replaced by a miniensemble of a pair of subclassifiers with a random linear oracle to choose between the two. It is argued that this approach encourages extra diversity in the ensemble while allowing for high accuracy of the individual ensemble members. Experiments were carried out with 35 data sets from UCI and 11 ensemble models. Each ensemble model was examined with and without the oracle. The results showed that *all* ensemble methods benefited from the new approach, most markedly so random subspace and bagging. A further experiment with seven real medical data sets demonstrates the validity of these findings outside the UCI data collection.

Index Terms—Classifier ensembles, fusion and selection, random hyperplane, multivariate (oblique) decision trees.

1 Introduction

Classifier fusion and classifier selection are two complementary approaches to designing classifier ensembles [18]. The underlying assumption in classifier fusion is that the classifiers have "expertise" across the whole feature space and are likely to misclassify different objects. To derive the class label for a new object **x**, the decisions of the classifiers in the ensemble are combined by a consensus-type rule, for example, majority voting. Conversely, in classifier selection, the classifiers are assumed to have complementary expertise. When a new object **x** is submitted for classification, a single "most competent" classifier is chosen and given the authority to assign the class label. Classifier selection assumes the existence of an oracle that selects the classifier with the highest competence for **x**.

In this study, we propose to combine selection and fusion within a single ensemble. To build each classifier, first, a random oracle is created in the form of a hyperplane. The data in each half-space is used to train a classifier within the chosen ensemble approach. During classification, the oracle for each classifier is applied, and the respective subclassifier makes the decision to be fused further at the ensemble level. The paper is organized as follows: Section 2 explains classifier selection. Section 3 details the proposed random linear oracle approach and gives a brief reference to multivariate decision trees. Section 4 explains why the random oracle idea works. The experimental details and results with 35 data sets from UCI are given in Section 5. Further results on seven real medical data sets confirm the findings beyond the UCI data collection. Section 7 concludes the study.

2 Classifier Selection

The idea of classifier selection resurfaced several times under different names in the past 30 years [1], [8], [16], [20], [31], [36]. The following approaches can be detailed [21]:

- **Static classifier selection**. The regions of competence of each classifier are specified during a training phase, prior to classifying. In the operation phase, an object **x** is submitted for classification. The region of **x** is first found, and the classifier responsible for this region is called upon to label **x** [2], [31], [39].
- **Dynamic classifier selection**. The choice of a classifier to label **x** is made during the classification. The classifier with the highest competence gives the label of **x**. The oracle here consists of estimating the accuracies (competences) and pronouncing the winner [9], [10], [13], [31], [34], [35], [37], [42]. The difference between the first and the second approaches reduces to whether or not evaluation of competence is carried out during the classification. Specifying the regions is, in fact, a prejudged competence and can be viewed as a faster version of the dynamic classifier selection approach.

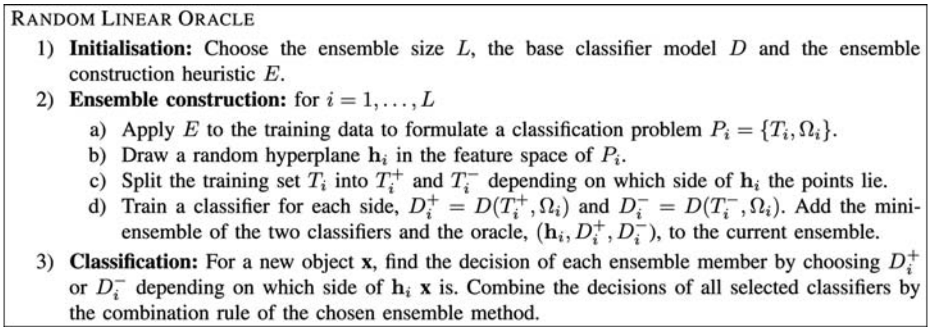
3 Random Linear Oracle

Switching between selection and fusion was proposed in [21]. If the dominance of the nominated classifier over the remaining classifiers is not statistically significant, the whole ensemble is summoned, and the classifier decisions are fused. Otherwise, the nominated classifier alone makes the decision. A natural fusion-selection scheme is the so-called mixture of experts [2], [16], [36]. The classifiers and the oracle are trained together so that the classifiers are pushed into specializing in different regions of the feature space, developed as part of the training. Along with

- *L.I. Kuncheva is with the School of Informatics, University of Wales, Bangor, Dean Street, Bangor, Gwynedd LL57 1UT, UK. E-mail: l.i.Kuncheva@bangor.ac.uk.*
- *J.J. Rodríguez is with the Escuela Politecnica Superior, Edificio C, Universidad de Burgos, c/ Francisco de Vitoria s/n. 09006 Burgos, Spain. E-mail: jjrodriguez@ubu.es.*

Manuscript received 5 July 2006; revised 13 Dec. 2006; accepted 19 Dec. 2006; published online 24 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0330-0706. Digital Object Identifier no. 10.1109/TKDE.2007.1016.

1041-4347/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society



```
RANDOM LINEAR ORACLE
1) Initialisation: Choose the ensemble size $L$, the base classifier model $D$ and the ensemble
   construction heuristic $E$.
2) Ensemble construction: for $i = 1, \dots, L$
    a) Apply $E$ to the training data to formulate a classification problem $P_i = \{T_i, \Omega_i\}$.
    b) Draw a random hyperplane $\mathbf{h}_i$ in the feature space of $P_i$.
    c) Split the training set $T_i$ into $T_i^+$ and $T_i^-$ depending on which side of $\mathbf{h}_i$ the points lie.
    d) Train a classifier for each side, $D_i^+ = D(T_i^+, \Omega_i)$ and $D_i^- = D(T_i^-, \Omega_i)$. Add the mini-
       ensemble of the two classifiers and the oracle, $(\mathbf{h}_i, D_i^+, D_i^-)$, to the current ensemble.
3) Classification: For a new object $\mathbf{x}$, find the decision of each ensemble member by choosing $D_i^+$
   or $D_i^-$ depending on which side of $\mathbf{h}_i$ $\mathbf{x}$ is. Combine the decisions of all selected classifiers by
   the combination rule of the chosen ensemble method.
```

Fig. 1. The generic algorithm for building a classifier ensemble with a random linear oracle.

enforcing this differentiation, the oracle learns which classifier to trust most for a given $\mathbf{x}$. The oracle in this case is able to assign weights of competence to the classifiers depending on the input $\mathbf{x}$ instead of choosing a single most competent classifier. Thus, the ensemble decision is derived as a fusion of weighted opinions. Data-dependent fusion has been advocated as a more accurate alternative of mere fusion [17], [28]. As usual, the success of flexible and powerful approaches such as these critically depends upon the availability of a bespoke training procedure.

This paper proposes a different fusion-selection scheme based on a random oracle. The idea is to replace each classifier in the ensemble by a miniensemble of two classifiers and an oracle, where the oracle is a random linear function. When a new object comes for classification, the oracle for the respective classifier decides which subclassifier to use. The labels issued by the subclassifiers are then combined through the ensemble combination rule. During training, the ensemble heuristic is applied first. For example, prior to the oracle stage, the training set may be selected by resampling or reweighting the data, feature subsets may be selected or extracted, or supraclasses may be formed within the Error Correcting Code (ECOC) ensemble approach. The random linear oracle approach is generic because it "encapsulates" only the base classifier and can fit within any ensemble strategy or base classifier model. Even more, the random oracle itself may serve as the ensemble-building heuristic.

Fig. 1 gives a formal description of the random linear oracle procedure for any chosen ensemble method. In the notations in the figure, a classification problem P is defined as a labeled training set T and a set of classes Ω. A classifier model (learner) $D(T, \Omega)$ is a training procedure to derive a classifier from a given labeled training set T with labels from Ω. An ensemble method is characterized by an ensemble heuristic E and a combination rule. For example, the Random Subspace ensemble method selects a random feature subset for each ensemble member. Thus, applying E to the training data to obtain classification problem P_i will return a set T_i with all the objects in the original training data but with a random subset of features. The set of classes Ω will be the same as the original set. For an ensemble using the ECOC method, E will return a training set T_i identical to the original training set, but the set of labels Ω_i will represent a two-class problem by a predefined grouping of the original classes. The ensemble construction framework with random oracle is laid out in a sequential way in Fig. 1 so that incremental ensemble methods such as AdaBoost can be accommodated. Even if E is the identity mapping, that is, it reproduces the original classification problem with training set T and class set Ω, the Random Oracle method can generate a viable ensemble.

The proposed model is different from the standard classifier selection paradigm where one oracle governs the whole ensemble. Multiple random oracles also make the proposed approach different from the mixture-of-experts model and the switching model discussed above.

The linear oracle approach touches upon an area that, at first glance, appears to be far from classifier selection—multivariate (or oblique) decision trees. Decision trees are termed *oblique* when the split at each node is not necessarily parallel to the feature axes. In classical decision tree induction, one feature is selected for each node, and the optimal split of the node into children nodes is determined so as to optimize a given criterion. Oblique trees may use any function of any subsets of features at each node. To gain from this flexibility, ingenious training algorithms are required. Oblique trees have been found to be much smaller and equally accurate compared to standard trees [6], [29]. Linear functions are the traditional choice. Perceptron-like algorithms have been proposed, whereby the coefficients of the hyperplane at each node are sequentially updated with the presentation of each new training example reaching that node [6]. The approaches vary from randomized hill climbing [30] to evolutionary algorithms [7], [38], simulated annealing [14], and tabu search [24]. The criterion being optimized at each node is usually the minimum classification error, but other criteria have also been proposed, for example, the squared error [6], the minimum message length [38], the classifiability [25], or impurity [30].

The difference between all these approaches and the *random* linear oracle is that the oracle is not supposed to optimize any criterion; the oracle merely serves as a divider of the space into two random halves. Any training of this hyperplane will be harmful because it will take away the intended diversity. In this line, all linear classifiers such as the Fisher's discriminant, Naïve Bayes, the logistic classifier, and others are not suitable as oracles. The most logical choice here seems to be a random split. It has been observed that just a few training iterations are sufficient to arrive at a near optimal hyperplane for a node in the tree [30]. Optimizing classification accuracy at each internal node is a greedy strategy whose overall optimality is not guaranteed. Thus,

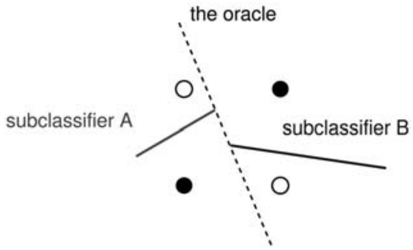


Fig. 2. XOR classification problem and its solution using a linear oracle and two linear subclassifiers.

the random oracle at the root node is not necessarily harmful with respect to the overall performance of the tree.

The proposed fusion-selection scheme can be recast as a classifier ensemble of the so-called "omnivariate trees" [25], [43]. In omnivariate decision trees, the function to be used at each node is not specified in advance and can be picked from a set of functions during the induction of the tree. In our case, all ensemble members will be omnivariate trees, where there will be a random linear function at the root node followed by standard univariate subtrees. In the rest of the paper, we will use the fusion-selection metaphor because it expresses better our motivation and results. Note that we do not use any of the training approaches for omnivariate or multivariate trees, so the analogy stops here.

4 Why Does Random Oracle Work?

Although empirical studies about classifier ensembles abound, theoretical results are still limited [18]. The reason for this is the complexity of ensemble models compared to single classifiers. Being a more versatile model than single classifiers, ensembles can learn the training data with a higher precision, but this is not a guarantee that they will fare better on new unseen data. Another difficulty in formalizing classifier ensemble methods comes from the fact that ensembles rely on the diversity between the members. Diversity itself is a controversial and difficult to formulate concept. Besides, its relationship with the ensemble accuracy is not straightforward [19], [22]. Therefore, we present two intuitive reasons to explain why random oracle may work.

The success of the random oracle idea can be attributed to two factors:

1. By splitting the feature space into two parts, the classification task may become easier for the chosen classifier model. Thus, the **individual accuracy** of the ensemble members is expected to be higher than or at least no worse than that of a "monolith" classifier over the whole feature space. This is similar in spirit to the divide-and-conquer strategy, whereby a problem is decomposed into subproblems that are (supposedly) easier to solve individually. Although expected, higher individual accuracy is not guaranteed by any means, as explained later.
2. The classification of a data point $\mathbf{x}$ will be made by one of the two subclassifiers of each ensemble member. Since the subclassifiers have been trained on very different data subsets (determined by the random oracle), **diversity** is expected to be large.

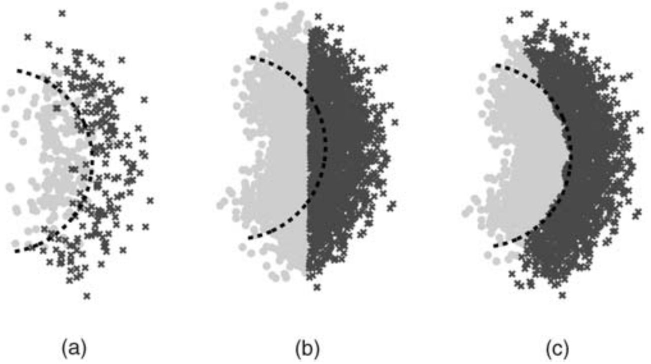


Fig. 3. (a) The training data set, (b) the testing data set labeled by the classical bagging ensemble, and (c) the testing data set labeled by the ensemble bagging + oracle. For reference, the optimal classification boundary is superimposed with a dashed line.

As a simple example illustrating the first factor, consider the XOR problem in Fig. 2. Suppose that the base classifier is linear. Clearly, the base classifier cannot provide a perfect separation of the two classes. However, any split of the data into two nonempty subsets will result in two linearly separable classes (one of these may contain a single point).

To demonstrate the diversity factor, we run an experiment with a synthetic data set. The training set of 400 points is plotted in Fig. 3a. Consider again a linear base classifier. Eleven ensemble members were built using different bootstrap samples of the data (standard bagging). Finally, bagging with linear oracle was applied. For each of the classifiers, two different random points were chosen from the training data set, and the perpendicular bisector of the segment between the two points was taken to be the hyperplane (the separating line in the 2D case). The two subclassifiers were trained on the points from the bootstrap sample falling on the two sides of the separating line. For testing, a separate data set of 4,000 points was generated from the distribution of the problem. The averaged individual error of the (linear) classifiers in the bagging ensemble was 0.2167, whereas it comes at no surprise that the averaged individual error for the classifiers with the oracle was substantially lower, 0.1380. The ensemble errors were 0.2168 for the classical bagging and 0.1212 for the ensemble with the oracle. Fig. 3b plots the testing data set as labeled by the classical bagging ensemble. This was done to visualize the classification boundary obtained through the ensemble. Fig. 3c shows the testing data labeled by the ensemble with the oracle. The shape of the ensemble boundary is far more adequate, which is also reflected in the ensemble accuracies.

Margineantu and Dietterich [26] devised the so-called "kappa-error" diagrams to show the relationship between the diversity and the individual accuracy of the classifiers. Plotted in a kappa-error diagram are $L(L-1)/2$ points, where L is the ensemble size. Each point corresponds to a pair of classifiers, say, D_1 and D_2. On the x-axis is a measure of diversity between the pair, kappa. Kappa evaluates the level of agreement between two classifier outputs while

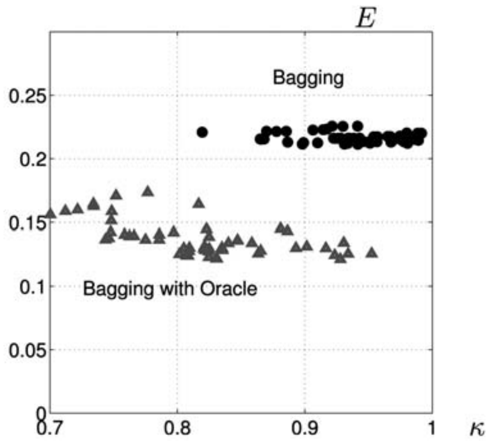


Fig. 4. A kappa-error diagram for the two ensembles: classical bagging and bagging with a random linear oracle.

correcting for chance [11]. For two classes, as in the above example, kappa is calculated as

$$\begin{aligned}\kappa_{i,j} = 2(m_{1,1}m_{2,2} - m_{1,2}m_{2,1})/ \\ ((m_{1,1} + m_{1,2})(m_{1,1} + m_{2,1}) \\ + (m_{1,2} + m_{2,2})(m_{2,1} + m_{2,2})),\end{aligned} \tag{1}$$

where $m_{k,s}$ is the proportion of the data set used for testing, which D_1 labels as ω_k and D_2 labels as ω_s. Low values of κ signify high disagreement and, hence, high diversity. If the classifiers produce identical class labels, $\kappa = 1$. Alternatively, if the classifiers are independent, $\kappa = 0$. Independence is not necessarily the best scenario in multiple classifier systems [23]. Even more desirable is "negative dependence," $\kappa < 0$, whereby classifiers commit related errors so that when one classifier is wrong, the other has a more than random chance of being correct.

On the y-axis of a kappa-error diagram is the averaged individual error of classifiers D_1 and D_2, $E_{1,2} = \frac{E_1+E_2}{2}$. As small values of κ indicate better diversity and small values of $E_{1,2}$ indicate better accuracy, the most desirable pairs of classifiers will lie in the bottom-left corner. Fig. 4 shows the kappa-error diagram for the two ensembles. The cluster of 55 points corresponding to all pairs in the bagging ensemble is higher than the cluster corresponding to the ensemble with the oracle. The individual accuracy of the members is markedly better for the ensemble with the oracle. Also, as expected, the cluster for the ensemble with the oracle is more to the left, showing lower kappa, hence, better diversity. It is worth mentioning that bagging is known to produce ensembles with relatively low diversity; this is why the values of kappa are close to 1. The situation is further aggravated by choosing a linear classifier as the base classifier. Being a stable classifier that does not vary much with small changes in the training data, the linear classifier is not well suited for bagging. We chose it here for illustration purposes only. The averaged diversity kappa across all pairs of classifiers for the bagging ensemble was 0.9376, whereas, for the ensemble with the oracle, it was 0.8153. Even though this difference between diversities might look insignificant at a first glance, it is likely to fetch noticeable improvement on the ensemble performance when combined with the high individual accuracy of the ensemble members.

The choice of points that determine the position of the hyperplane is random. In the worst case, there will be one point or a very small number of points on one side of the hyperplane, and all the other points will lie together on the other side. In this case, the benefit from the oracle vanishes, as practically one classifier is responsible for the whole training data. Thus, the ensemble member with oracle is reduced to an ordinary ensemble member, which is not going to cause a big drop on the overall ensemble accuracy. The small cutoff of points may not be adequate for training the corresponding classifier well. However, assuming that the training set represents the population of interest well, only a negligible number of points will have to be labeled by that classifier. Thus, the overall number of errors will not increase dramatically.

The sacrifice made by the oracle-based ensemble approach is that the training data is split into two, so one of the subclassifiers in the pair is always trained on less than half of the training data. This will add instability to the trained classifier that, however, transfers into extra diversity in the ensemble. On the other hand, as only a part of the feature space is presented to the classifier, the problem may be easier to solve, therefore requiring a smaller training sample anyway.

5 Experiments

The goal of this experiment is to find out whether the Random Linear Oracle makes any difference to the accuracy of standard ensemble methods.

A summary of the 35 data sets from UCI [3] used in this study is given in Table 1. To calculate the hyperplane, each categorical feature for each data set was replaced by C binary features, where C is the number of possible categories. For example, a feature with three categories, "a," "b," and "c" is represented by three binary features x_a, x_b, and x_c, respectively. If the value for a particular object is "c," then for this object, $x_a = 0$, $x_b = 0$, and $x_c = 1$. All numerical features were linearly scaled within the interval [0, 1] using the minimum and maximum value in the training data.

Decision trees have been used as the base classifier. They are invariant with respect to scaling the features and also handle categorical features by multivariate splits. Thus, the transformation of the categorical features into binary and the scaling of the numerical features were needed only for the hyperplane and for determining on which side of it a given point lies. The only exception here is the Rotation Forest ensemble that relies on extracting linear features and hence needs all the data in a numerical format.

With each data set, 10 tenfold cross validations were carried out using Weka [41]. The ensemble methods selected for the experiment are listed in alphabetical order in Table 2. All ensemble methods were run on the same training-testing splits with and without the Random Linear Oracle. The testing accuracy was recorded for each method on each data set. In this way, 100 estimates of the testing accuracy were available for each method and each data set, suitable for paired tests as well.

TABLE 1
Summary of the 35 UCI Data Sets Used in the Experiment

Data set	Classes	Objects	D	C
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	2	303	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	2	294	7	6
hypothyroid	4	3772	22	7
ionosphere	2	351	0	34
iris	3	150	0	4
kr-vs-kp	2	3196	36	0
labor	2	57	8	8
letter	26	20000	0	16
lymphography	4	148	15	3
mushroom	2	8124	22	0
pima-diabetes	2	768	0	8
primary-tumor	22	339	17	0
segment	7	2310	0	19
sick	2	3772	22	7
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-context	11	990	2	10
vowel-nocontext	11	990	0	10
waveform	3	5000	0	40
wisconsin-bc	2	699	0	9
zoo	7	101	16	2

"D" stands for the number of discrete features and "C" for the number of continuous-valued features.

TABLE 2
Ensemble Methods

Name	Source	Details
AdaBoost.M1 (S)	[12]	Re-sampling version
AdaBoost.M1 (W)	[12]	Re-weighting version
Bagging	[4]	Bootstrap samples
Decorate	[27]	Incremental method with artificially constructed examples to enhance diversity
Ensemble^a	–	The only ensemble heuristic is the Random Linear Oracle
Multiboost (S)	[40]	Re-sampling version
Multiboost (W)	[40]	Re-weighting version
Random Subspace (50%)	[15]	Random subsets of features, 50% selected for each classifier
Random Subspace (75%)	[15]	Random subsets of features, 75% selected for each classifier
Random Forest^b	[5]	Ensemble of randomised decision trees
Rotation Forest	[32]	Random PCA-based sparse rotation of the feature space

1) All methods except ^{a} and ^{b} appear in four versions in the experiment: with pruned trees (notation "-P-" is used in the sequel), with unpruned trees ("-U-"), with oracle ("H" for hyperplane), and without oracle ("N"). 2) ^{a} The Ensemble method does not have a nonoracle version, so two versions are considered: H-P-Ensemble and H-U-Ensemble. 3) ^{b} Random Forest uses a special unpruned randomized decision tree; therefore, the two versions used here are H-U-Random Forest and N-U-Random Forest. 4) There are 40 ensemble methods altogether.

The base classifier model D in all experiments was a standard decision tree (J48), except for Random Forest, which uses bespoke randomized trees. Both pruned and unpruned trees were considered, as there is no consensus as to which strategy is better for ensembles.

The hyperplane $\mathbf{h}_i$ was generated by taking a random pair of points from the training set T_i and calculating the hyperplane perpendicular to the line segment between the points and running through the middle point. In this way, we made sure that there were points on both sides of $\mathbf{h}_i$.

A summary of the experimental results is given in Table 3. The ensemble methods are sorted by their overall ranks. To calculate the rank of a method, the mean classification accuracies of all methods are sorted for each data set. The method with the best accuracy receives rank 1, the second best receives rank 2, and so on. If there is a tie, the ranks are shared. For example, if the second, third, and fourth best accuracies are the same, all three methods receive rank 3. For each method, there are 35 rank values, one for each data set. The overall rank of a method is the averaged rank of this method across the 35 data sets. The smaller the rank, the better the method. The overall ranks are also shown in the table.

Differences between the averaged ranks may be due not only to the random oracle but also to the advantages of the ensemble methods over one another. We want to find out whether the random oracle has the desired effect. The Win-Tie-Loss column in Table 3 gives the number of data sets for which the method with the Random Linear Oracle has been better-same-worse compared to the method *without* the oracle. To find out the statistical significance of the difference, we carry out a sign test on wins, losses, and ties as explained in [33]. If the oracle and the nonoracle versions of the ensemble methods are equivalent, each method will be expected to win on approximately half of the data sets. For a relatively large number of data sets n, the number of wins follows a normal distribution with mean $\frac{n}{2}$ and standard deviation $\frac{\sqrt{n}}{2}$. The critical value is $n_c = \left\lceil \frac{n}{2} + z_\alpha \frac{\sqrt{n}}{2} \right\rceil$, where z_α is the z-value for the specified level of significance α, and $\lceil \cdot \rceil$ denotes "ceiling." Any result where the number of "wins" plus half of the number of the

TABLE 3
UCI Data: Ensemble Methods with and without the Random Linear Oracle Sorted by Their Average Ranks

Method	Total Rank	Win-tie -loss	Benefit	Method	Total Rank	Win-tie -loss	Benefit
H-P-Rotation Forest	10.83	15-2-18	▌	N-U-Rand. Subs. (50%)	20.74	–	
H-U-Rotation Forest	11.01	15-3-17	▌	N-P-AdaBoostM1 (W)	21.07	–	
N-P-Rotation Forest	11.16	–		N-P-AdaBoostM1 (S)	21.20	–	
N-U-Rotation Forest	11.70	–		N-U-MultiBoost (S)	21.30	–	
H-U-Rand. Subs. (50%)	15.40	• 26-2-7	█████	H-U-Bagging	21.41	• 28-1-6	███████
H-U-Rand. Subs. (75%)	16.47	• 32-1-2	████████████	N-U-Random Forest	21.44	–	
H-P-MultiBoost (W)	16.49	21-0-14	█	H-U-AdaBoostM1 (W)	21.47	20-0-15	▌
H-P-MultiBoost (S)	16.87	• 24-1-10	██	N-U-AdaBoostM1 (W)	22.13	–	
N-P-MultiBoost (W)	17.74	–		N-U-AdaBoostM1 (S)	22.79	–	
H-P-Rand. Subs. (75%)	17.83	• 32-1-2	██████████	N-P-Rand. Subs. (50%)	23.29	–	
H-U-Random Forest	18.26	21-1-13	███	H-P-Decorate	23.30	19-1-15	▌
H-P-Rand. Subs. (50%)	18.63	• 25-1-9	█████	N-P-Decorate	23.80	–	
H-U-MultiBoost (W)	18.63	• 23-2-10	██	H-P-Ensemble	24.66	N/A	
H-U-MultiBoost (S)	18.80	• 24-0-11	██	H-U-Decorate	25.10	• 25-2-8	███
H-P-AdaBoostM1 (S)	19.20	22-2-11	██	H-U-Ensemble	26.63	N/A	
N-P-MultiBoost (S)	19.31	–		N-P-Bagging	26.96	–	
H-P-Bagging	20.36	• 30-1-4	███████	N-P-Rand. Subs. (75%)	27.69	–	
H-U-AdaBoostM1 (S)	20.46	23-0-12	██	N-U-Bagging	28.04	–	
H-P-AdaBoostM1 (W)	20.54	20-0-15	▌	N-U-Decorate	28.06	–	
N-U-MultiBoost (W)	20.61	–		N-U-Rand. Subs. (75%)	28.63	–	

"H" (for hyperplane) indicates that the oracle is present, "N" indicates the standard version without the oracle, "-P-" is for ensemble with pruned trees, and "-U-" is for ensembles with unpruned trees.

ties is greater than or equal to n_c indicates a statistically significant difference. For $\alpha = 0.05$ and $n = 35$ data sets, $n_c = \lceil 17.5 + 1.96 \times \sqrt{35}/2 \rceil = 24$. The methods for which the random oracle approach leads to a statistically significant improvement are marked with a bullet in Table 3. There is no method where the random oracle led to significantly worse results.

The results in Table 3 show a consistent tendency. With no exception, the ensemble method with the Random Linear Oracle has a total rank better than the total rank without the oracle.

Of the 19 ensemble methods in total, only Rotation Forest has the number of wins with oracle lower than the number of losses. Nevertheless, the oracle improves the general performance so that Rotation Forest with the oracle is ranked higher than without the oracle. This finding, illogical at first glance, can be explained by the following argument. The 15 wins have been achieved by a larger margin in the ranks compared to the 18 (17) losses. The sum of ranks therefore slightly favors the oracle version of the method.

The Random Linear Oracle by itself is not a sufficiently viable ensemble heuristic. The two ensembles based solely on the oracle H-P-Ensemble and H-U-Ensemble (with pruned and unpruned trees, respectively) have low total ranks. To evaluate which ensemble methods benefit the most from the oracle, the column "Benefit" in Table 3 displays the gain in rank scores for the 19 ensemble methods. The length of the bar corresponds to the rank difference between the version with oracle ("H-") and the standard method (without oracle, "N-"). The two ensemble approaches that benefit the most are random subspace and bagging. Both are simple nonincremental approaches to which the random oracle induces some welcome additional diversity. The results indicate that ensemble approaches that are based on engineered diversity, for example, boosting, benefit less, regardless of their rating with respect to other ensembles. One possible reason for this is that introducing the oracle upsets the well-measured ensemble construction procedure, and the extra randomization renders itself redundant. Finally, there is no clear pattern as to whether the oracle favors pruned or unpruned trees.

The random linear oracle approach does not increase substantially the computational complexity of the ensemble methods. In the training stage, two subclassifiers need to be trained instead of one classifier for each ensemble member. However, each subclassifier only uses part of the training data T_i. If the data size is a factor in the training complexity, then the random oracle may, in fact, be faster to train. In the classification stage, calculation of each classifier's decision is preceded by a linear calculation of the score on the hyperplane $\mathbf{h}_i$, which will not cause a great delay. The ensemble size is the same as the ensemble without the oracle, only, instead of univariate trees, the ensemble consists of a fixed type of omnivariate trees as discussed above.

6 Further Experiments with Seven Medical Data Sets

Finally, to verify the above results outside the UCI data collection, we repeated the experiment, with the same protocol, on seven real medical data sets explained in Table 4.[1] This selection is intended as a sample from a specific class of data sets characterized by 1) a small number of true classes, which may or may not correspond to coherent clusters; 2) a moderate number of observations (up to a few hundreds); and 3) a moderate number of features

[1] Available for download at http://www.informatics.bangor.ac.uk/~kuncheva/activities/patrec1.html.

TABLE 4
Summary of the Seven Real Medical Data Sets

Data set	Classes	Objects	D	C	Comment
Weaning	2	302	0	17	Courtesy of Dr. A.Temelkov, M.D. Centre of Acute Respiratory Insufficiency, Alexandrov's University Hospital, Sofia, Bulgaria
Respiratory	2	85	5	12	Courtesy of Dr. N. Jekova, M.D. Neonatal Clinic, University Hospital "Maichin Dom", Sofia, Bulgaria
Laryngeal-1	2	213	0	16	Courtesy of Dr. D. Doskov, M.D. Phoniatrics Department, University Hospital "Queen Joanna", Sofia, Bulgaria
Laryngeal-2	2	692	0	16	Courtesy of Dr. Stefan Hadjitodorov Central Laboratory of Biomedical Engineering Bulgarian Academy of Sciences, Sofia, Bulgaria
Laryngeal-3	3	353	0	16	(as Laryngeal 2)
Voice-3	3	238	1	9	(as Laryngeal 1)
Voice-9	9	428	1	9	(as Laryngeal 1)

"D" stands for the number of discrete features and "C" for the number of continuous-valued features.

TABLE 5
Medical Data: Ensemble Methods with and without the Random Linear Oracle Sorted by Their Average Ranks

Method	Total Rank	Win-tie -loss	Benefit	Method	Total Rank	Win-tie -loss	Benefit
H-U-Rand. Subs. (50%)	6.14	6-0-1	█████████	N-U-Decorate	20.29	–	
H-P-Rand. Subs. (50%)	8.29	7-0-0	██████████	H-U-AdaBoostM1 (S)	20.43	4-0-3	██
H-P-Rotation Forest	8.50	4-0-3	████	H-P-Bagging	21.93	6-0-1	███████
H-U-Rotation Forest	8.57	6-0-1	█████	N-U-AdaBoostM1 (S)	22.14	–	
H-U-Random Forest	9.36	3-1-3	██	N-U-MultiBoost (S)	22.36	–	
N-U-Random Forest	11.29	–		N-P-AdaBoostM1 (W)	22.43	–	
N-P-Rotation Forest	12.21	–		N-P-MultiBoost (W)	22.86	–	
H-U-MultiBoost (W)	12.93	5-0-2	█████	H-U-Rand. Subs. (75%)	23.36	7-0-0	██████████████
N-U-Rotation Forest	13.36	–		N-U-AdaBoostM1 (W)	23.86	–	
N-U-Rand. Subs. (50%)	15.00	–		H-P-Rand. Subs. (75%)	24.50	7-0-0	██████████
H-P-MultiBoost (S)	16.14	5-0-2	████	H-P-AdaBoostM1 (W)	25.43	3-0-4	negative
H-P-MultiBoost (W)	16.43	6-0-1	███████	H-P-Decorate	26.43	3-0-4	zero
H-U-AdaBoostM1 (W)	16.86	6-0-1	███████	N-P-Decorate	26.43	–	
H-P-AdaBoostM1 (S)	17.14	5-0-2	███████████	N-U-Bagging	27.64	–	
N-U-MultiBoost (W)	18.14	–		N-P-AdaBoostM1 (S)	28.29	–	
N-P-Rand. Subs. (50%)	18.29	–		N-P-Bagging	28.71	–	
H-U-Decorate	19.43	5-0-2	█	N-P-Rand. Subs. (75%)	34.71	–	
H-U-MultiBoost (S)	19.57	3-0-4	███	H-P-Ensemble	36.29	–	
N-P-MultiBoost (S)	20.14	–		H-U-Ensemble	37.00	–	
H-U-Bagging	20.14	6-0-1	████████	N-U-Rand. Subs. (75%)	37.00	–	

"H" (for hyperplane) indicates that the oracle is present, "N" indicates the standard version without the oracle, "-P-" is for ensemble with pruned trees, and "-U-" is for ensembles with unpruned trees.

(typically 5 to 30). Such data sets are often collected, for example, in clinical medicine for pilot research studies.

The results are displayed in Table 5. There are statistically significant differences between all the ensemble methods ($p \approx 0$) by the Friedman's analysis of variance (ANOVA) for the ranks. Since the number of data sets in this verification experiment is small, consistency of the results may be expected to drop. All 7-0-0 patterns of Win-Draw-Loss in Table 5 indicate a statistically significant improvement at $p < 0.05$ of the oracle ensemble over the same ensemble model without the oracle. Patterns of 6-0-1 indicate significance at $p < 0.1$. Even for this small number of data sets, most ensembles will show better performance with oracle than without oracle. In many cases, the benefit from the oracle (represented by the length of the black box) is even larger compared to that with the 35 UCI data sets.

With the exception of H-P-AdaboostM1 (W) and H-P-Decorate, in all other 17 cases, the hyperplane oracle improves on the ensemble without the oracle.

7 Conclusion

We propose a combined fusion-selection approach to classifier ensemble design, which we call the Random Linear Oracle. Each classifier in the ensemble is replaced by a miniensemble of a pair of subclassifiers with an oracle to choose between them. The oracle is in the form of a hyperplane, randomly drawn and fixed for each ensemble member. The results with 35 data sets and 20 ensemble models, each one with and without the oracle, show that all ensemble methods benefited from the new approach, albeit in different degrees. The oracle was most useful for the random subspace and bagging ensembles. The results were further verified, and the findings were confirmed on seven real medical data sets.

In this study, we chose the simplest random oracle: the linear one. There is no reason why we should stop here. Different split functions may work better for some ensemble models or data types. It is also interesting to try a different model of the base classifier, for example, Naïve Bayes or a neural network, again with all ensemble models, with and without the oracle. The explanation in Section 4 on why random oracle works is not tied up with either the choice of the split function or the base classifier model. Hence, the proposed fusion-selection approach is expected to work regardless of the specific choices.

However, we are cautious to extend our claim to all types of problems. There are interesting and complex problems out there that are still a challenge to pattern recognition and machine learning communities. For example, Knowledge Discovery and Data Mining (KDD) competitions have set a high standard over the years by putting up such thought-provoking problems. Bespoke methods have been developed to address large data sizes, the subtleties of text mining and Internet retrieval, heavily imbalanced classes, and so forth. These methods may not work well for more standard data. Our proposed ensemble method is not meant to address all types of challenges, and we recognize that it might not be superior to the same competitors in a different scenario.

References

[1] E. Alpaydin and M.I. Jordan, “Local Linear Perceptrons for Classification,” *IEEE Trans. Neural Networks,* vol. 7, no. 3, pp. 788-792, May 1996.

[2] R. Avnimelech and N. Intrator, “Boosted Mixture of Experts: An Ensemble Learning Scheme,” *Neural Computation,* vol. 11, pp. 475-490, 1999.

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

[4] L. Breiman, “Bagging Predictors,” Technical Report 421, Dept. of Statistics, Univ. of California, Berkeley, 1994.

[5] L. Breiman, “Random Forests,” *Machine Learning,* vol. 45, pp. 5-32, 2001.

[6] C.E. Brodley and P.E. Utgoff, “Multivariate Decision Trees,” *Machine Learning,* vol. 19, no. 1, pp. 45-77, 1995.

[7] E. Cantú-Paz and C. Kamath, “Inducing Oblique Decision Trees with Evolutionary Algorithms,” *IEEE Trans. Evolutionary Computing,* vol. 7, no. 1, pp. 54-68, Feb. 2003.

[8] B.V. Dasarathy and B.V. Sheela, “A Composite Classifier System Design: Concepts and Methodology,” *Proc. IEEE,* vol. 67, pp. 708-713, 1978.

[9] L. Didaci and G. Giacinto, “Dynamic Classifier Selection by Adaptive k-Nearest-Neighbourhood Rule,” *Proc. Fifth Int'l Workshop Multiple Classifier Systems (MCS '04),* pp. 174-183, 2004.

[10] L. Didaci, G. Giacinto, F. Roli, and G.L. Marcialis, “A Study on the Performances of Dynamic Classifier Selection Based on Local Accuracy Estimation,” *Pattern Recognition,* vol. 38, no. 11, pp. 2188-2191, 2005.

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

[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] G. Giacinto and F. Roli, “An Approach to the Automatic Design of Multiple Classifier Systems,” *Pattern Recognition Letters,* vol. 22, pp. 25-33, 2001.

[14] D.G. Heath, S. Kasif, and S. Salzberg, “Induction of Oblique Decision Trees,” *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI),* pp. 1002-1007, 1993.

[15] T.K. Ho, “The Random Space Method for Constructing Decision Forests,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 832-844, Aug. 1998.

[16] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, “Adaptive Mixtures of Local Experts,” *Neural Computation,* vol. 3, pp. 79-87, 1991.

[17] M.S. Kamel and N.M. Wanas, “Data Dependence in Combining Classifiers,” *Proc. Fourth Int'l Workshop Multiple Classifier Systems (MCS '03),* pp. 1-14, 2003.

[18] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms.* John Wiley & Sons, 2004.

[19] L.I. Kuncheva and C.J. Whitaker, “Measures of Diversity in Classifier Ensembles,” *Machine Learning,* vol. 51, pp. 181-207, 2003.

[20] L.I. Kuncheva, “Change-Glasses Approach in Pattern Recognition,” *Pattern Recognition Letters,* vol. 14, pp. 619-623, 1993.

[21] L.I. Kuncheva, “Switching between Selection and Fusion in Combining Classifiers: An Experiment,” *IEEE Trans. Systems, Man, and Cybernetics, Part B,* vol. 32, no. 2, pp. 146-156, Apr. 2002.

[22] L.I. Kuncheva, “Diversity in Multiple Classifier Systems (Editorial),” *Information Fusion,* vol. 6, no. 1, pp. 3-4, 2005.

[23] 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.

[24] X.B. Li, J.R. Sweigar, J.T.C. Teng, J.M. Donohue, L.A. Thombs, and S.M. Wang, “Multivariate Decision Trees Using Linear Discriminants and Tabu Search,” *IEEE Trans. Systems, Man, and Cybernetics, Part A,* vol. 33, no. 2, pp. 194-205, Mar. 2003.

[25] Y. Li, M. Dong, and R. Kothari, “Classifiability-Based Omnivariate Decision Trees,” *IEEE Trans. Neural Networks,* vol. 16, no. 6, pp. 1547-1560, Nov. 2005.

[26] D.D. Margineantu and T.G. Dietterich, “Pruning Adaptive Boosting,” *Proc. 14th Int'l Conf. Machine Learning,* pp. 378-387, 1997.

[27] P. Melville and R.J. Mooney, “Creating Diversity in Ensembles Using Artificial Data,” *Information Fusion,* vol. 6, no. 1, pp. 99-111, 2005.

[28] P. Moerland and E. Mayoraz, “Dynaboost: Combining Boosted Hypotheses in a Dynamic Way,” Technical Report IDIAP-RR99-09, Dalle Molle Inst. for Perceptual Artificial Intelligence (IDIAP), 1999.

[29] K.V.S. Murthy, “On Growing Better Decision Trees from Data,” PhD dissertation, Johns Hopkins Univ., 1995.

[30] S.K. Murthy, S. Kasif, and S. Salzberg, “A System for Induction of Oblique Decision Trees,” *J. Artificial Intelligence Research,* vol. 2, pp. 1-32, 1994.

[31] L.A. Rastrigin and R.H. Erenstein, *Method of Collective Recognition,* 1981 (in Russian).

[32] J.J. Rodríguez, L.I. Kuncheva, and C.J. Alonso, “Rotation Forest: A New Classifier Ensemble Method,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 10, pp. 1619-1630, Oct. 2006.

[33] J. Demšar, “Statistical Comparison of Classifiers over Multiple Data Sets,” *J. Machine Learning Research,* vol. 7, pp. 1-30, 2006.

[34] H.W. Shin and S.Y. Sohn, “Selected Tree Classifier Combination Based on Both Accuracy and Error Diversity,” *Pattern Recognition,* vol. 38, no. 2, pp. 191-197, 2005.

[35] S. Singh and M. Singh, “A Dynamic Classifier Selection and Combination Approach to Image Region Labelling,” *Signal Processing—Image Comm.,* vol. 20, no. 3, pp. 219-231, 2005.

[36] F. Smieja, “The Pandemonium System of Reflective Agents,” *IEEE Trans. Neural Networks,* vol. 7, no. 1, pp. 97-106, Jan. 1996.

[37] P.C. Smits, “Multiple Classifier Systems for Supervised Remote Sensing Image Classification Based on Dynamic Classifier Selection,” *IEEE Trans. Geoscience and Remote Sensing,* vol. 40, no. 4, pp. 801-813, Apr. 2002.

[38] P.J. Tan and D.L. Dowe, “MML Inference of Oblique Decision Trees,” *Proc. 17th Australian Joint Conf. Artificial Intelligence (AI '04),* pp. 1082-1088, 2004.

[39] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, and A. Gelzinis, “Soft Combination of Neural Classifiers: A Comparative Study,” *Pattern Recognition Letters,* vol. 20, pp. 429-444, 1999.

[40] G.I. Webb, “MultiBoosting: A Technique for Combining Boosting and Wagging,” *Machine Learning,* vol. 40, no. 2, pp. 159-196, 2000.

[41] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques,* second ed. Morgan Kaufmann, 2005.

[42] K. Woods, W.P. Kegelmeyer, and K. Bowyer, “Combination of Multiple Classifiers Using Local Accuracy Estimates,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, pp. 405-410, 1997.

[43] O.T. Yildiz and E. Alpaydin, “Omnivariate Decision Trees,” *IEEE Trans. Neural Networks,* vol. 12, no. 6, pp. 1539-1546, Nov. 2001.

Ludmila I. Kuncheva received the MSc degree from the Technical University, Sofia, Bulgaria, 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 Electronics and Computer Science, University of Wales, Bangor, United Kingdom. Her research interests include pattern recognition and classification, machine learning, classifier combination, and nearest neighbor classifiers. She is a member of the IEEE.

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 in the Department of Computer Science, University of Valladolid, from 1995 to 2000. Currently, he is working in 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.