Random Projection Random Discretization Ensembles—Ensembles of Linear Multivariate Decision Trees

Amir Ahmad and Gavin Brown

Abstract—In this paper, we present a novel ensemble method *random projection random discretization ensembles* (RPRDE) to create ensembles of linear multivariate decision trees by using a univariate decision tree algorithm. The present method combines the better computational complexity of a univariate decision tree algorithm with the better representational power of linear multivariate decision trees. We develop random discretization (RD) method that creates random discretized features from continuous features. Random projection (RP) is used to create new features that are linear combinations of original features. A new dataset is created by augmenting discretized features (created by using RD) with features created by using RP. Each decision tree of a RPRD ensemble is trained on one dataset from the pool of these datasets by using a univariate decision tree algorithm. As these multivariate decision trees (because of features created by RP) have more representational power than univariate decision trees, we expect accurate decision trees in the ensemble. Diverse training datasets ensure diverse decision trees in the ensemble. We study the performance of RPRDE against other popular ensemble techniques using C4.5 tree as the base classifier. RPRDE matches or outperforms other popular ensemble methods. Experiments results also suggest that the proposed method is quite robust to the class noise.

Index Terms—Ensembles, decision trees, discretization, randomization, random projections, noise

1 Introduction

Ensembles are a combination of multiple base models [1]–[3]; the final classification depends on the combined outputs of individual models. Classifier ensembles have shown to produce better results than single models, provided the classifiers are *accurate* and *diverse* [2]. Ensembles perform best when base models are *unstable*–classifiers whose output undergoes significant changes in generalisation with small changes in the training data-decision trees and neural networks are in this class.

Several methods have been proposed to build decision tree ensembles. Randomization is introduced to build diverse decision trees. *Bagging* [4] and *Boosting* [5] introduce randomization by manipulating the training data supplied to each classifier. Multiboosting [6] combines the principle of Bagging with AdaBoost. Ho [7] proposed *Random Subspaces* that selects random subsets of input features for training. Ho [7] suggested that random subspace method works well when there is certain redundancy in the dataset, especially in the collections of features. They suggest that for the random subspace method to work on datasets having a small number of features, redundancy need to be introduced artificially by using simple functions of the features. Breiman [8] combined Random Subspaces technique with Bagging to create *Random Forests*. To build a tree, it uses a bootstrap replica of the training sample, then during the tree growing phase, at each node the optimal split is selected from a random subset of size *K* of candidate features. Rodriguez *et al.* [9] proposed *Rotation Forests*, producing new input features using Principal Component Analysis (PCA). Dietterich [10] proposed a method to grow ensembles that consist of randomized trees; instead of selecting the best split, they select a test uniformly among the best *K* tests. Geurts *et al.* [11] proposed *Extremely Randomized Trees* (ERT), *Extremely Randomized Trees* combines the feature randomization of Random Subspaces with a totally random selection of the cut-point. Random Decision Trees (RDT) [12], [13] proposed by Fan *et al.* use completely random splits points.

As for large ensembles, the significant difference between different ensemble methods almost disappear [14], there has been research to develop strategies that can create small ensembles with good or near optimal performance [15]. Small ensembles have advantage of faster prediction process.

In this work, we develop a novel ensemble method *Random Projection Discretization Ensembles* (RPRDE) that is a combination of popular random projection (RP) method [16], [17] and the proposed Random Discretization (RD) method. RPRDE, with a univariate decision tree algorithm, creates the ensembles of multivariate decision trees. This kind of mechanism to create decision tree ensembles has not been used in any popular ensemble method. Hence,

- *A. Ahmad is with the Faculty of Computing and Information Technology, King Abdulaziz University, Rabigh 21911, Saudi Arabia. E-mail: amirahmad01@gmail.com.*
- *G. Brown is with the School of Computer Science, University of Manchester, Manchester, M13 9PL, U.K. E-mail: gavin.brown@manchester.ac.uk.*

Manuscript received 8 June 2012; revised 13 June 2013; accepted 4 July 2013. Date of publication 4 Aug. 2013; date of current version 7 May 2014. Recommended for acceptance by L. Khan.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TKDE.2013.134

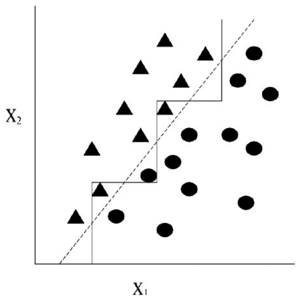

Fig. 1. Example of univariate (solid line) and linear multivariate (dashed line) splits that separate instances of two classes in two dimensional data.

it can be combined with any popular ensemble method. One of the motivations for this method is that it should work well with small ensembles. Creating accurate decision trees is the key for these kinds of ensembles, we will discuss later that RPRD trees are likely to be quite accurate.

In Section 2, we discuss linear multivariate decision trees. In Section 3, we present the principle of RP and its application for tree ensembles. In Section 4, we introduce RD. Section 5 deals with the motivation for combining RP and RD. Experiments and the performance evaluation are presented in Section 6. Section 7 describes conclusion and future work.

2 Linear Multivariate Decision Trees

Decision trees are built using top down induction methods. In a decision tree, each node partitions the available patterns into two sets. Univariate decision trees like C4.5 [18] are limited to testing a single feature at a node. This reduces the representational power of decision tree as univariate decision trees are restricted to splits of the instance space that are orthogonal to the feature's axis. Multivariate decision trees [19], [20] overcome the representational limitation of univariate decision trees (Fig. 1). Linear multivariate decision trees allow the node to test a linear combination of the numeric features. This test can be presented as,

$$s = \sum_{i=1}^{m} c_i x_i \leq Z, \tag{1}$$

where x_i are the numeric features, c_i are the corresponding real valued coefficients, m the number of attributes and Z is a numeric constant. These trees are also called oblique decision trees as they uses oblique (non-axis-parallel) hyperplanes to partition the data. In some problem domains multivariate decision trees perform better than univariate decision trees [21]. However, these trees are not very popular as it is computationally expensive to create these trees [22].

Different approaches have been proposed to create linear multivariate decision trees. Breiman *et al.* [19] suggested a method to create multivariate decision trees that uses a perturbation algorithm. SADT [22] uses simulated annealing to compute hyperplanes. Simulated annealing introduces an element of randomness so SADT generates a different decision tree in each run. OC1 [23] improves SADT by combining deterministic hill climbing with randomization.

In the next section, we discuss random projection (RP) as a technique to construct diverse linear multivariate decision trees.

3 Random Projection (RP) Ensembles

In this section, we discuss random projection and its application to create diverse linear multivariate decision trees. For many data mining techniques, it is difficult to handle high dimensional data. Dimensionality reduction is a method to handle high dimensional data. Principal Component Analysis (PCA) is a popular choice for the dimensionality reduction problem. However, it is quite expensive to compute for high dimensional data. Random projection (RP) has been emerged as a powerful method for the dimensionality reduction problem. Random projection is a technique of mapping a number of points in a high-dimensional space into a low dimensional space with the property that the Euclidean distance of any two points is approximately preserved through the projection. The key idea behind RP is Johnson and Linden Strauss theorem [16], [17] that states that if the points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances and relative angles between the points [24] are approximately preserved.

In RP, the original data is projected onto a lower dimensionality subspace using a random matrix whose columns have unit lengths. Using matrix notation where $D_{m\times n}$ is the original set of n, m dimensional observations. The projection of the data onto a lower d-dimensional subspace is defined as

$$D^{RP}_{d\times n} = R_{d\times m} D_{m\times n}, \tag{2}$$

where $R_{d\times m}$ is a Random Matrix and $D^{RP}_{d\times n}$ is the new $d \times n$ projected matrix.

RP has been successfully used for creating clustering ensembles as different random projections create different clustering results [25]. Schclar and Rokach [26] introduced a classifier ensemble method by using random projections. In this method, new datasets were created by using random projections. Classifiers learn on those new diverse datasets were diverse, hence diverse classifiers were created. They used nearest-neighbour classifier as the base classifier to show that their method outperformed the Bagging algorithm. However, no comparative study with more complex ensemble methods like AdaBoost.M1 was presented. In the present work, we use RP for following reasons.

1) **Datasets created by using random projection have features that are linear combinations of original features -** RP projects the original data to a new feature space. These features are linear combinations of original features. If a univariate decision tree is trained on the projected data, we get a orthogonal decision tree for the projected data. This tree is an oblique tree in the original feature space. The main difference between this approach of creating oblique decision trees and other approaches discussed in Section 2 is that in this approach the orientation of a hyperplane is fixed (that depends on new features), only the location of the hyperplane

```
Input- Dataset $T$ with $m$ continuous features and $k$ classes $(c_1, c_2, .., c_k)$. $L$ the size of the ensemble.
Training Phase
for i=1...$L$ do
    Data Generation
    Use the Random Projection $RP_i$ to generate dataset $T_i$ consisting of $d$ new features.
    Learning Phase
    Learn $D_i$ decision tree on $T_i$.
end for
Classification Phase
For a given data point $\mathbf{x}$
for i=1...$L$ do
    Convert $\mathbf{x}$ into a $d$ dimensional data point $\mathbf{x_i}$ by using the Random Projection $RP_i$.
    Let $p_{i,j}(\mathbf{x})$ be the probability for $\mathbf{x_i}$ by the decision tree $D_i$ to the hypothesis that $\mathbf{x}$ comes from class $c_j$. Calculate $p_{i,j}(\mathbf{x})$ for all classes (j = 1..k).
end for
Calculate the confidence $C(j)$ for each class $c_j$ (j = 1..$k$) by the average contribution method, $C(j) = \frac{1}{L}\sum_{i=1}^{L} p_{i,j}(\mathbf{x})$.
The class with the largest confidence will be the class of $\mathbf{x}$.
```

Fig. 2. Algorithm for RP ensembles.

is learned during the tree growing phase whereas in the other approaches both location and orientation of the hyperplane are computed during the tree growing phase.

2) **Different random projections create different datasets, hence, diverse decision trees (that learn on these datasets) are obtained -** To build an ensemble of decision trees, we need diverse decision trees. RP helps in creating these diverse decision trees. Different random projections of a dataset create different new datasets. If we train univariate decision trees on these new datasets, we get diverse decision trees that are linear multivariate decision trees in the original feature space. We can combine these trees to get an ensemble of decision trees.

The algorithm for an ensemble of RP decision trees is given in Fig. 2. In the next section, we describe Random Discretization (RD) method that can be used to create decision tree ensembles.

4 Random Discretization (RD) Ensembles

In this section, we present Random Discretization ensemble method. We also discuss the representational power of RD ensembles. First, we describe a novel method, Randomized Discretization (RD), that creates diverse discretized datasets.

Discretization divides the features values into different categories depending upon intervals they fall into. For example, if we want to discretize a feature into three categories, we need two points $x1$ and $x2$ between the maximum (x_{max}) and the minimum (x_{min}) values of the feature, if $x1 < x2$ a features value x is discretized using following rules;

if ($x \leq x1$) the category of x = 1,
if ($x > x1$) and ($x \leq x2$) the category of x = 2,
if ($x > x2$) the category of x = 3,
where x is the feature value.

```
Random_Discretization
Input- Numeric training dataset T with $n$ data points and $m$ continuous features.
Output- Discretized training data set.
Begin
1- For $s$ categories in each dimension, select $s$-1 data points randomly from the training data.
for j=1...$m$ do
    2- Get the $j^{th}$ features values of $s$-1 points and sort them.
    3- If all points are equal to the minimum or the maximum values of the feature, select $s$-1 points randomly having values between the minimum or the maximum values of the feature.
end for
for i=1...$n$ do
    3- Discretize the $i^{th}$ data point using the values got in step 2.
end for
End.
```

Fig. 3. Random Discretization (RD) method.

To create s categories, we need $s-1$ points. There are different methods to create these points. However, all these methods produce a single and unique discretized dataset. For creating ensembles, we need diverse datasets so that the learning on these datasets creates diverse classifiers.

We propose a novel method Random Discretization (RD) to select these $s-1$ points. In this method, we introduce randomization in discretization of various features. To create s categories $s-1$ *data points are selected randomly from the training data*. For each feature, every selected data point has one value, this way we can have $s-1$ data points for every feature. The feature can be discretized into s categories using these $s-1$ data points. It is possible that for some features we have less than $s-1$ boundaries as two or more selected data points may have same values for these features. That will produce less number of categories for those features. In the extreme case for some features, all selected points have features values equal to the minimum or the maximum values of the features. In other words, we have no point for the features between the minimum and the maximum values of the feature. $s-1$ points are selected randomly between the minimum and the maximum values of the feature for these kinds of features. The proposed RD algorithm is presented in Fig. 3.

Table 1 shows a two dimensional dataset. For dividing every dimension into three categories, we need two points. Two data points are selected randomly, for example we select D2 and D4. For the feature X, 3.7 and 6.8 are the two points for the discretization. For the feature Y, 5.8 and

TABLE 1
Example Dataset

Data point	feature X	feature Y	Class
D1	2.3	6.8	1
D2	3.7	5.8	2
D3	3.8	9.8	1
D4	6.8	4.5	1
D5	3.2	8.1	2
D6	7.7	4.2	2

```
Input- Dataset $T$ with $m$ continuous features and $k$ classes $(c_1, c_2, .., c_k)$. $L$ the size of the ensemble.
Training Phase
for i=1...$L$ do
    Data Generation
    Use Random Discretization $RD_i$ to generate integer valued dataset $T_i$.
    Learning Phase
    Treat dataset $T_i$ as continuous and learn $D_i$ decision tree on it.
end for
Classification Phase
For a given data point $\mathbf{x}$
for i=1...$L$ do
    Convert $\mathbf{x}$ into a discretized data point $\mathbf{x_i}$ by using Random Discretization $RD_i$.
    Let $p_{i,j}(\mathbf{x})$ be the probability for $\mathbf{x_i}$ by the decision tree $D_i$ to the hypothesis that $\mathbf{x}$ comes from class $c_j$. Calculate $p_{i,j}(\mathbf{x})$ for all classes (j = 1..k).
end for
Calculate the confidence $C(j)$ for each class $c_j$ (j = 1..$k$) by the average contribution method.
$C(j) = \frac{1}{L}\sum_{i=1}^{L} p_{i,j}(\mathbf{x})$.
The class with the largest confidence will be the class of $\mathbf{x}$.
```

Fig. 4. Algorithm for RD ensembles.

4.5 are the two points for the discretization so the dataset is discretized by using following rules,

1) **For X feature**
   if X feature value$\leq$ 3.7, the category of X = 1,
   if 6.8 $\geq$ X feature value $>$ 3.7, the category of X = 2,
   if X feature value $>$ 6.8, the category of X = 3.
2) **For Y feature**
   if Y feature value $\leq$ 4.5, the category of Y = 1,
   if if 5.8 $\geq$ Y feature value $>$ 4.5, the category of Y = 2,
   if Y feature value>5.8, the category of Y = 3.

Each decision tree in an ensemble learns on one discretized dataset from the pool of different datasets created by RD. If the order of values of discretized features is maintained, the discretized dataset is an ordinal data. A decision tree treats this dataset as continuous during the learning phase. Results of different decision trees in the ensemble are combined to get the final prediction. The algorithm for RD ensembles is presented in Fig. 4.

4.1 Motivation for Random Discretization (RD) Ensembles

In this section, we focus our discussion on C4.5 type decision trees (univariate decision trees). In an ensemble, we need accurate and diverse classifiers. RD builds an ensemble of classifiers by changing category boundaries. Univariate decision trees have representational problem because of their orthogonal properties. They have difficulty in learning non-orthogonal decision boundaries with limited amount of data. We will discuss in this section that when we have infinite RD decision trees in an ensemble, *a piece-wise continuous function produced by the RD ensemble approximates to a diagonal concept (non-orthogonal concept) for a two dimensional data*. Hence, RD ensembles have better representational power as compared to a single decision tree.

Ensembles of diverse decision trees solve the representational problem associated with a single univariate decision tree as combined results of decision trees produce a good approximation of a non-orthogonal concept. Dietterich [1] shows that for majority voting an ensemble of small size decision trees is similar to a large size decision tree and can create a good approximation of a diagonal concept. To get diverse decision trees, we need trees with different split points. In RD, the dataset is discretized randomly. When decision trees learn on the discretized datasets, nodes can split only on the bin boundaries. As the number of boundaries is small (for s categories, there are only s-1 category boundaries) and these boundaries are random, there is a small probability that decision trees trained on different discretized datasets have same node splits for a feature. *In other words, there is a high probability that each decision tree divides the data space at different points*. Hence, it creates diverse decision trees. In case of infinite RD decision trees in an ensemble, a piece-wise continuous function produced by RD ensembles approximates to the diagonal concept. We may extend this argument to the other decision boundaries to show that RD ensembles have good representational power. On the basis of the above argument, it can be hypothesized that a RD ensemble has better representational power as compared to a single decision tree.

The success of ensemble methods depends on the creation of uncorrelated classifiers [3]. A RD tree has limited tree growing options as it has to follow bin boundaries. In other words, diverse decision trees are produced as different options are provided (different bin boundaries) at tree growing phase. Very accurate classifiers may not be obtained by using the RD method, however, this technique ensures very diverse decision trees with good representation powers for RD ensembles. In the next section, we present the RPRDE method that combines the RP transformation and the RD transformation.

5 Random Projection Random Discretization Ensembles (RPRDE)

In the last two sections, we discussed two different ensemble methods; RP ensembles and RD ensembles. In this section, we discuss our proposed approach RPRDE that combines features created by RP and RD methods.

Different ensemble methods have been proposed. Some of them are based on different mechanisms (see Section 1) like Bagging [4], AdaBoost.M1[27] and Random Subspaces [7] etc. whereas some of the ensemble methods combine methods that have different mechanisms, for example Random Forests [8] combines Bagging with Random Subspaces, Multiboosting [6] combines Bagging with AdaBoost and Rotation Forest [9] combines randomization in the feature space division with Bagging. The basic idea behind these "hybrid" ensemble techniques is that as the mechanisms differ for different ensemble methods, their combination may out-perform either in isolation.

RD and RP have different mechanisms; RD is based on random discretization in input space whereas RP creates new features that are the linear combinations of the original features. We propose that they can be combined to create better ensemble method (RPRDE). In RD, we create m discretized features, whereas RP creates d features (d < m)

```
Input- Original dataset $T$ with $m$ continuous features and $k$ classes $(c_1, c_2, .., c_k)$.
$L$, the size of the ensemble.
Training Phase
for i=1...$L$ do
    Data Generation
    1- Use Random Discretization $RD_i$ to create $m$ discretized features $S_i$.
    2- Use Random Projection $RP_i$ to create $d$ features $R_i$.
    3- Combine $S_i$ and $R_i$ features to get $m + d$ dimensional dataset $T_i$.
    Learning Phase
    Treating dataset $T_i$ as continuous, learn $D_i$ decision tree on it.
end for
Classification Phase
For a given data point $\mathbf{x}$
for i=1...$L$ do
    1- Convert $\mathbf{x}$ into $m + d$ dimensional data point $\mathbf{x_i}$ using Random Discretization ($RD_i$) and
    Random Projection ($RP_i$).
    2- Let $p_{i,j}(\mathbf{x})$ be the probability for $\mathbf{x_i}$ by the decision tree $D_i$ to the hypothesis that $\mathbf{x}$
    comes from class $c_j$. Calculate $p_{i,j}(\mathbf{x})$ for all classes (j = 1..$k$).
end for
Calculate the confidence $C(j)$ for each class $c_j$ (j = 1..$k$) by the average contribution method,
$C(j) = \frac{1}{L}\sum_{i=1}^{L} p_{i,j}(\mathbf{x})$.
The class with the largest confidence will be the class of $\mathbf{x}$.
```

Fig. 5. Algorithm for the RPRDE method.

(these d features are the linear combinations of the original m features). We combine these two feature spaces to get $m + d$ features. A univariate decision tree is trained on this $m+d$ dimensional data. Though we train the univariate decision tree, we get decision surfaces both orthogonal (due to m discretized features) and oblique to the axes defined by the features of the input space (due to the new d features). *We expect that RPRD trees have more representational power as compared to RD decision trees and RP decision trees as they have more decision surfaces*.

Decision trees do the feature selection at each node [18]. This property is useful for these type of datasets (when the original features are augmented with new features), as new features will only be selected when they are better features for classification than the original features. Hence, trees trained on the new data (RP features + RD features) may be better classifiers as compared to decision trees with the original features.

Accurate decision trees with reasonable diversity are useful for small ensembles [9]. RPRD trees use all the data points of the training dataset (for example in Bagging and AdaBoost, a tree use a part of the training data, in Random Subspaces a tree uses some of the features). RPRD trees have more representational power as new features are added to the original features (these are discretized features so we loss some of the information). We do not employ any randomization process during the tree growing process as in Random Forests and ERT. We expect accurate decision trees because of these reasons. Good accuracy of RPRD trees is useful for the success of small RPRD ensembles.

New training datasets are created by using different RD transformations and RP transformations, hence, diverse datasets are obtained. Decision trees trained on these diverse datasets are expected to be diverse. The algorithm to create a RPRD ensemble is presented in Fig. 5.

5.1 The Analysis of RPRDE

As discussed in Section 2, univariate decision trees have representational problem. An ensemble of decision trees has better representational power. In Section 4, we discussed the learning of a diagonal problem by RD ensembles. The question arises whether RD ensembles or any other decision tree ensembles techniques can learn a diagonal decision boundary problem accurately. Ahmad *et al.* [28] showed that infinite-sized ensembles, consisting of finite sized decision trees, created by a pure randomized method (split points are created randomly), cannot learn a diagonal problem accurately. This suggests that these methods can be improved upon for some sets of problems by changing the decision boundaries. In RPRDE, we are including RP features which are linear combinations of the original features. The decision boundary will change in this new feature space. There is a high likelihood that in this new feature space, decision tree ensembles can learn the problem easily. We could not show theoretically that RPRDE can learn each decision boundary accurately. There is a possibility that like Random Forests [29], RPRDE is not consistent. In the next section, we will discuss the structure of RPRD tree to understand the behaviour of RDPDE.

5.2 The Structure of RPRD Trees

To understand the performance of RPRD ensembles, we also carried out experiments in which features created using RP were combined with the original features (In RPRDE, RP features are concatenated with RD features). The performance of ensembles created using the method was not comparable to RPRD ensembles. The following argument suggests that the good combination of RP features and RD features in a RPRD tree is the reason for the success of RPRD ensembles.

Experiments suggest that C4.5 trees do not perform well with random projections [30]. Fradkin [30] suggests "*Random projections and decision trees are perhaps not a good combination*". This means new features created by using random projections are not as informative as the original features. Hence, when we combine the original features with the features created using random projections and train a univariate decision tree on it, there is a strong probability, that the original features are selected at higher levels as they are more informative, whereas the features created by using random projections will be selected at lower levels as they are less informative. This suggests that these trees are not very diverse (as they are similar at higher levels).

In RPRD trees, the discretized original attributes are used. These attributes are different for different trees (they are created with RD features), hence even if these attributes are selected at higher levels, they create diverse trees. As there is a loss of information due to the discretization, it makes the original attributes less informative. Hence, when the discretized attributes are used, then there is more probability that attributes created by using random projections will be selected at higher levels of decision trees. That ensures more diverse trees as different trees use different new attributes (different trees use different attributes created using different random projections).

5.3 Weaknesses

Both RD and RP can be applied only for the pure continuous datasets. That restricts the application of RPRDE. In this approach, we use random projections to create new features that add extra computational cost. These new features are added to the original features that increase the size of the training dataset. Hence, the tree learning phase may need more computational resources as compared to the training phase for the data with the original dataset. However, the performance of RPRDE justifies the additional computational cost. In the next section, we present experimental results.

5.4 The Related Work

In RPRD new attributes (RD attributes and RP attributes) are created and combined to create new datasets. In this section, we will discuss some related works.

In one of the variants of Random Forests [8], Breiman defined new features by taking random linear combinations of features. This is done to introduce more diversity among decision trees. In rotation forests [9] new input features are created by using Principal Component Analysis (PCA). RS [7] is good when there is certain redundancy in features. For datasets, where there is no redundancy, redundancy needs to be introduced artificially by concatenating new features that are linear combinations of original features to the original features and treating this as the data.

Balcan *et al.* [31] suggested a mapping that uses a given kernel and random unlabelled examples. *This mapping generates a set of features such that if a dataset is linearly separable with a margin under the kernel, then it is approximately separable in this new feature space.* Augmenting kernel features with original features has been suggested by Balcan and Blum [32]. Rwebangira [33] observed by using different types of classifiers that if these new kernel features are concatenated with the original features then the classifiers trained on these new datasets perform better than the classifiers trained on the original data or the data created by kernel features. These results suggest that classifiers can use strengths of both kinds of features, this results in the improved performance of classifiers [33]. However, this proposal is only for single classifier and in RPRDE discretized features (created from original features) are used instead of original features to increase the diversity of the decision trees.

6 Experiments

We carried out the comparative study of RPRDE method against the other popular ensemble methods by using Weka package [34] and the Dice softwate [12] (http://www.dice4dm.com/) to test the effectiveness of RPRDE approach. In this section, we present our experimental results.

We carried out experiments with Bagging [4], Adaboost.M1 [5], Multiboosting [6], and Random Forests [9] modules from Weka. For RDT [12], we used Dice software. As discussed in Section 1, Bagging and RF create diverse trees by using random methods. Adaboost.M1 is a boosting algorithm and Multiboosting combines Bagging and Adaboost. Therefore, in our comparative study, we have compared our method with different kinds of decision tree ensemble methods. For RD ensembles, RP ensembles, RPRDE, and Bagging, unpruned J48 (Weka implementation of C4.5 [18]) was used. Whereas, for Adaboost.M1 and Multiboosting, we carried out experiments with pruned J48 and unpruned J48. For pruned decision tree, default confidence threshold (0.25) for pruning was used. For Random Forests, the number of features selected to select from at each node is set at $\lfloor \log_2(|m| + 1) \rfloor$ (default value). We carried out experiments on two different sizes of ensembles. The sizes were selected such that one may represent small ensembles and the other large ensembles. There is no clear definition of small ensembles and large ensembles. Following [9], for small ensembles we selected the size of an ensemble as 10. Different sizes of ensembles were selected in different papers, following [11] we selected the size of ensembles as 100 for large ensembles.

For Multiboosting, when the ensemble size was 10, the number of subcommittees was chosen as 3 whereas for the ensemble size 100, the number of subcommittees was chosen as 10. Default settings were used for the rest of the parameters. Datasets were normalized (linear scaling to the unit range) to bring all the features on the same scale.

The experiments were conducted following the 5×2 cross-validation [35]. The original test, proposed by Dietterich [35], to compare the performance of classifiers suffers from low replicability. Alpaydin [36] proposes a modification to the 5×2 cross-validation F test. We used this test for our experiments. We considered a confidence level of 95% for this test. (P) with an ensemble method in the paper represents that the base classifier is unpruned J48, whereas (U) represents that the base classifier is unpruned J48.

6.1 Parameters for RPRD

There are three parameters in RPRDE,

1) **Number of bins** - In RD, the data is discretized by creating bins. If the number of bins is large we get discretized data similar to the original data. If the number of bins is small there is a large loss of the information. We conducted preliminary experiments with 2-10 bins and found that for different datasets, the best classification results were obtained at different number of bins, however it was in the range of 3-6. **Hence, in the experiments for all the datasets, 5 bins were created.** For each tree in an ensemble, different 4 points from the training dataset were selected randomly to create 5 bins.
2) **Dimension d of the datasets created using RP** - Fern and Brodley [25] suggest *"to our knowledge it is still an open question how to choose the dimensionality for a random projection in order to preserve separation among clusters in general clustering algorithms."* However, Dasgupta [37] shows that the data from a mixture of K Gaussians can be projected into just $O(\log K)$ dimensions while retaining the approximate level of separation between clusters.

TABLE 2
Average Classification Errors (in %) on the Simulated Data with a Diagonal Concept

Size	RPRDE	RD ensembles	RP ensembles	Bagging	AdaBoost (U)	AdaBoost (P)	Multi-boosting(U)	Multi-boosting(P)	Random Forests	RDT	Single Tree
10	**8.01**	8.93(+)	10.65(+)	10.44(+)	10.16(+)	10.85(+)	9.68(+)	10.67(+)	11.07(+)	18.14(+)	15.76(+)
100	**4.92**	6.41(+)	5.09	9.12(+)	6.95(+)	7.71(+)	6.31(+)	7.21(+)	8.30(+)	7.25(+)	15.76(+)

Bold numbers show the best performance. '+/-' shows that the performance of RPRDE is statistically better/worse (by using the statistical test proposed in [36]) than that algorithm. RPRDE ensembles perform similar to or better than other ensemble methods. P represents that the base classifier is unpruned J48, whereas U represents that the base classifier is unpruned J48.

We selected d as 2($\log_2 m$) where m is the number of features. As in almost all datasets we tested, number of classes (k) < number of features (m) and it was assumed that each class probability distribution was represented by a Gaussian distribution. Factor 2 was taken to have a large dimension of the projected data so that the most of the information was preserved. There was no guarantee that with this assumption, the correct value of d was obtained. However, these new d features were added to original features (discretized features) so even if these new d features did not have all the information of the dataset, their combination with original features might improve the representational power of decision trees.

3) **Matrix for Random Projection -** The elements r_{ij} of Random Matrix R are often Gaussian distributed. Achlioptas [38] has shown that Gaussian distribution can be replaced by a distribution such as
$r_{ij} = \sqrt{3}\times(\pm 1)$ with probability 1/6 each, or 0 with 2/3 probability
We used this matrix for RP in our experiments as it has benefit of being easy to implement and compute.

For fair comparative study, no attempt was made to select the best parameters for different datasets. In summary, all the experiments were carried out with following parameters;

- The number of bin was 5.
- The number of new features created by using RP was 2($\log_2 m$), where m is the number of features.
- We used the matrix for RP as suggested by Achlioptas [38] (discussed above) in our experiments.

6.2 Controlled Experiments

As discussed in Section 4, univariate decision trees like C4.5 have difficulty in learning a diagonal concept. This experiment was carried to study the performance of RPRD ensembles for learning a diagonal concept. A 10 dimensional data having a 5 dimensional diagonal concept was created for this purpose. i^{th} feature of the data point x was defined by x_i, (i =1 to 10) where x_i is a random number between 0 to 1. Two classes were defined as

$$\sum_{i=1}^{5} x_i \leq 5/2, \tag{3}$$

and

$$\sum_{i=1}^{5} x_i > 5/2. \tag{4}$$

2000 data points were created. Experiments were done by using 5×2 cross-validation. Results are presented in Table 2. Results indicate that RPRD ensembles perform statistically better than other popular ensemble methods (only the RP ensemble was similar to RPRDE when the ensemble size was 100). This shows that RPRD ensembles can learn diagonal concepts very well. This vindicates our hypothesis that RPRD ensembles have good representational power.

6.3 Comparative Study

In the second part of the experiments, we selected 21 pure continuous data sets from the UCI Machine Learning Repository [39]. The information about datasets is presented in Table 3.

For the ensemble size 10, results are presented in Table 4. RPRDE is statistically similar to or better than other popular ensemble methods (Bagging (11 Wins/10 Draws), AdaBoost.M1(U) (8 Wins/13 Draws), AdaBoost.M1(P) (8 Wins/13 Draws), Multiboosting(U) (9 Wins/12 Draws), Multiboosting(P) (8 Wins/13 Draws), Random Forests (9 Wins/12 Draws) and RDT(17 Wins/4 Draws)). For the ensemble size 100, results are presented in Table 5. In this case, generally RPRD ensembles perform similar or better than the other popular ensembles methods, however the comparative advantage of RPRDE decreases (Bagging (13 Wins/8 Draws), AdaBoost.M1(U) (7 Wins/13 Draws/1 Loss), AdaBoost.M1(P) (7 Wins/13 Draws/1 Loss), Multiboosting(U) (6 Wins/14 Draws/1 Loss), Multiboosting(P) (7 Wins/13 Draws/1 Loss), Random Forests (7 Wins/14 Draws) and RDT (12 Wins/9 Draws)).

Experimental results suggest that RPRDE is comparatively better for small ensembles. This is true with the ensembles consisting of accurate classifiers with reasonable diversity [9]. We will discuss diversity and accuracy properties of RPRD ensembles in (Subsection 6.5).

RPRDE is the combination of two data transformation methods RP and RD. Our motivation to combine these two approaches is that they are based on different mechanisms so their combination may produce good results. Results indicate that for almost all the dataset (except Ring-Norm data, RP ensembles performed statistically better than RPRDE for the ensemble size 100) RPRDE is statistically better than both of these methods or similar to the

TABLE 3
Datasets Used in Experiments

Dataset Name	Size	No. of Classes	No. of attributes
Balance	625	3	4
Breast Cancer	699	2	9
Ecoli	336	8	7
Glass	215	7	9
Ionosphere	351	2	34
Letter	20000	26	16
Optical	5620	10	64
Pendigit	10992	10	16
Pima-diabetes	768	2	8
Phoneme	5404	2	5
RingNorm	7400	2	20
Satimage	6435	6	36
Segment	2310	7	19
Sonar	208	2	60
Spam	4601	2	57
TwoNorm	7400	2	20
Vehicle	846	4	18
Vowel	990	11	10
Waveform21	5000	3	21
Waveform40	5000	3	40
Yeast	1484	10	8

These datasets are pure datasets.

better one. For example, for the ensemble size 10, for Letter, Phoneme, Pendigit and Segment datasets, RPRDE is statistically better than both the methods, whereas for Optical, Spambase and Waveform40 datasets RPRDE is similar to RD ensemble method and better than RP ensemble method, and for the Ring-Norm dataset, RPRDE is similar to RP ensemble method and better than RD ensemble method. This behaviour suggests that RPRDE ensembles have got best of both methods.

6.4 Noisy Data

As some of the real datasets have class noise, it is important to study the robustness of RPRD ensembles for noisy data. The performance of boosting methods degrade in the presence of the class noise. Dietterich [10] tested this effect by introducing artificial noise in the class labels. AdaBoost has difficulty in learning when the dataset is noisy. In each iteration, the weight assigned to noisy data points increases so in subsequent iteration it concentrates more on noisy data points, it leads to overfitting of data. In this section, we present our experimental results to study the sensitivity of RPRDE to the class noise.

To add noise to the class labels, we followed the method of Dietterich [10]. To add classification noise at a rate r, we chose a fraction r of the instances and changed their class labels to be incorrect choosing uniformly from the set of incorrect labels. We carried out this exercise for noise levels 10% for all the datasets.

Results, when the size of the ensemble was 10, were presented in Table 6. Results indicates that RPRDE is quite robust to class noise. Their comparative advantage increases (as compared to without noise data) for noisy data. Except Bagging (performance of RPRDE is statistically worse than Bagging for Spambase data) RPRDE performed statistically similar or better than other popular ensemble methods (Bagging (12 Wins/8 Draws/1 Loss), AdaBoost.M1(U) (16 Wins/5 Draws), AdaBoost.M1(P) (16 Wins/5 Draws), Multiboosting(U) (15 Wins/6 Draws), Multiboosting(P) (14 Wins/7 Draws), Random Forests(14 Wins/7 Draws) and RDT(19 Wins/ 2 Draws)).

We also carried out same experiments when the size of ensemble was 100. Results are presented in Table 7. RPRDE performed statistically similar to or better than other popular ensembles methods (Bagging (12 Wins/9 Draws), AdaBoost.M1(U) (14 Wins/7 Draws), AdaBoost.M1(P) (14 Wins/7 Draws), Multiboosting(U) (12 Wins/9 Draws), Multiboosting(P) (12 Wins/9 Draws), Random Forests(9 Wins/12 Draws) and RDT (20 Wins/1 Draw)), however, its comparative advantage decreases (as compared to ensemble of size 10).

Results demonstrate that RPRD ensembles are quite robust to the class noise. This is due to the fact that RPRDE does not put so much emphasis on incorrectly classified instances as AdaBoost.M1 does. Overfitting is one of the weaknesses of oblique decision trees [21]. As RPRDE is using random methods (RP and RD) to create datasets, RPRDE avoids overfitting problem because of this. We are creating ensembles of RPRD trees this also helps in avoiding overfitting problem that is associated with single oblique decision tree. RDT is based on a pure randomized process, however, RDT did not perform well for the noisy data. This is an interesting observation and need further investigation. However, this is beyond the scope of this paper.

6.5 The Study of Ensemble Diversity

Kappa-error plots [40] is a method to understand the diversity-error behaviour of an ensemble These plots represent a point for each pair of classifiers in the ensemble. The x coordinate is a measure of diversity of the two classifiers D_i and D_j known as the *kappa* (κ) measure (low values suggest high diversity). The y coordinate is the average error $E_{i,j}$ of the two classifiers D_i and D_j. When the agreement of the two classifiers equals that expected by chance, $\kappa = 0$; when they agree on every instance, $\kappa = 1$. Negative values of κ mean a systematic disagreement between the two classifiers.

We draw kappa-error plots for five datasets (Pen, Phoneme, Segment, Vowel, Waveform21) for different ensemble methods. The scales of κ and $E_{i,j}$ are same for each given dataset so we can easily compare different ensemble methods. We plotted results of one of the testing phase of 5×2 cross-validation. The size of the ensembles was 10, so the total number of points was 45 in each plot. Plots are presented in Fig. 6. RPRDE is not as diverse

TABLE 4
Average Classification Errors (in %) for Different Ensembles Methods on Different Dataset for the Ensemble Size 10, Bold Numbers Show the Best Performance

Data	RPRDE	RD ensembles	RP ensembles	Bagging	AdaBoost (U)	AdaBoost (P)	Multi-boosting(U)	Multi-boosting(P)	Random Forests	RDT	Single tree
Balance	**13.41**	15.04(+)	14.88(+)	18.40(+)	20.51(+)	20.68(+)	19.30(+)	18.98(+)	19.75(+)	22.67(+)	21.95(+)
Breast Cancer	**3.41**	3.67	**3.41**	4.64	4.04	4.98	4.23	4.94	4.12	4.34	5.87(+)
Ecoli	16.13	16.72	**14.99**	18.45	18.24	17.12	18.09	17.62	18.45	57.27(+)	20.65
Glass	29.34	29.35	32.34	31.12	31.04	29.83	29.90	31.96	**26.54**	37.65(+)	34.02(+)
Ionosphere	7.06	7.26	**6.04**	7.47	7.92	7.74	9.57	9.33	7.52	16.73(+)	10.48(+)
Letter	**6.56**	7.32(+)	13.32(+)	9.94(+)	6.63	6.68	7.78(+)	7.72(+)	7.94(+)	41.31(+)	15.47(+)
Optical	3.48	3.77	5.20(+)	5.71(+)	**3.22**	3.38	3.90	4.26	4.17	7.53(+)	11.03(+)
Pendigits	**1.06**	1.41(+)	1.50(+)	2.85(+)	1.37(+)	1.40(+)	1.74(+)	1.94(+)	1.53(+)	9.17(+)	4.69(+)
Phoneme	**11.40**	14.54(+)	14.86(+)	13.04(+)	12.37	12.29	12.45	12.29	11.99	25.06(+)	15.63(+)
Pima-diabetes	24.89	25.39	24.97	**24.82**	27.08	29.01	26.01	25.78	25.72	30.22	26.89
Ring-Norm	3.24	4.41(+)	**3.23**	6.04(+)	4.14(+)	4.13(+)	4.91(+)	4.60(+)	5.84(+)	8.73(+)	10.10(+)
Satimage	**10.38**	10.70	11.02	11.38	10.67	10.48	10.88	10.56	10.62	12.02(+)	15.39(+)
Segment	**2.68**	3.36(+)	3.42(+)	4.11(+)	2.71	2.73	3.23(+)	3.52(+)	3.28(+)	8.35(+)	4.51(+)
Sonar	**21.54**	21.63	23.37	24.61	24.90	23.67	24.13	24.52	23.87	24.12	27.88(+)
Spambase	5.92	5.94	10.03(+)	6.13	5.53	5.51	**5.31**	5.44	5.99	7.53(+)	8.16(+)
Two-Norm	**3.71**	5.73(+)	4.28(+)	6.23(+)	5.90(+)	5.97(+)	6.16(+)	5.95(+)	6.23(+)	5.23(+)	16.16(+)
Vehicle	26.70	26.55	30.71	27.07	26.33	**25.53**	27.03	26.22	27.30	30.12	30.36
Vowel	**11.39**	12.87	13.23	20.49(+)	15.45(+)	15.73(+)	18.56(+)	17.53(+)	15.31(+)	48.27(+)	30.24(+)
Waveform21	16.99	17.52	**16.82**	18.67(+)	18.86(+)	18.97(+)	18.32(+)	18.20(+)	18.50(+)	18.73(+)	24.45(+)
Waveform40	**17.70**	18.11	20.26(+)	18.92(+)	19.24(+)	19.06(+)	18.66(+)	17.90	19.17(+)	23.67(+)	25.40(+)
Yeast	**42.06**	43.60	42.78	42.21	44.92(+)	45.71(+)	42.65	43.72	43.45	56.68(+)	47.25(+)
Win/Draw/Loss		7/14/0	8/13/0	11/10/0	8/13/0	8/13/0	9/12/0	8/13/0	9/12/0	17/4/0	18/3/0

'+/-' shows that the performance of RPRDE is statistically better/worse (by using the statistical test proposed in [36]) than that algorithm for that dataset. RPRD ensembles perform similar to or better than other ensemble methods. P represents that the base classifier is unpruned J48, whereas U represents that the base classifier is unpruned J48.

TABLE 5
Average Classification Errors (in %) for Different Ensembles Methods on Different Dataset for the Ensemble Size 100, Bold Numbers Show the Best Performance

Data	RPRDE	RD ensembles	RP ensembles	Bagging	AdaBoost (U)	AdaBoost (P)	Multi-boosting(U)	Multi-boosting(P)	Random Forests	RDT	Single tree
Balance	11.30	13.56	**11.26**	18.18(+)	21.86(+)	21.92(+)	20.35(+)	19.83(+)	18.91(+)	17.29(+)	21.95(+)
Breast Cancer	3.23	3.38	**2.98**	4.69(+)	3.61	3.89	3.49	4.10	3.89	3.87	5.87(+)
Ecoli	15.06	**14.40**	14.48	16.79	16.49	17.05	16.40	16.46	16.79	53.67(+)	20.65
Glass	27.85	27.29	30.00	28.88	27.57	28.84	28.60	28.72	**24.67**	33.14	34.02(+)
Ionosphere	5.81	6.56	**5.53**	6.89	7.86	6.48	7.79	6.68	6.50	12.83(+)	10.48(+)
Letter	4.22	4.93(+)	6.39(+)	8.52(+)	**4.08**	4.19	4.10	4.46	4.92(+)	22.58(+)	15.47(+)
Optical	2.45	2.76	2.11	4.60(+)	1.91(-)	**1.85**(-)	2.06	2.12	2.11	2.78	11.03(+)
Pendigits	**0.73**	0.93(+)	0.95(+)	2.29(+)	0.83	0.81	0.91(+)	0.89(+)	1.08(+)	5.06(+)	4.70(+)
Phoneme	10.54	13.96(+)	13.86(+)	12.21(+)	**10.48**	10.57	10.50	10.51	10.61	23.72(+)	15.63(+)
Pima-diabetes	23.57	23.62	25.47	**23.36**	26.20(+)	27.26(+)	25.21	25.79	23.88	28.91(+)	26.85
Ring-Norm	1.90	2.61(+)	**1.55**(-)	5.30(+)	2.27(+)	2.34(+)	2.44(+)	2.37(+)	4.45(+)	3.91(+)	9.82(+)
Satimage	9.11	9.47	9.90(+)	10.40(+)	8.91	**8.67**	9.02	8.73	9.21	10.72(+)	15.39(+)
Segment	**2.07**	2.77(+)	2.82(+)	3.70(+)	2.13	2.23	2.40	2.43	2.55(+)	6.35(+)	4.51(+)
Sonar	18.85	19.90	**18.17**	22.50	20.58	18.93	22.01	18.68	19.04	18.85	27.88(+)
Spambase	5.45	5.59	8.54(+)	5.92	5.29	4.99	**4.42**(-)	4.53(-)	5.06	5.91	8.16(+)
Two-Norm	**2.47**	3.64(+)	**2.47**	3.69(+)	2.83(+)	2.77(+)	2.87(+)	2.83(+)	3.65(+)	2.72	16.16(+)
Vehicle	24.42	24.97	29.43	26.87	23.86	**23.34**	24.04	24.80	25.90	29.24	30.36
Vowel	7.23	8.67	**7.17**	17.23(+)	10.77(+)	10.36(+)	11.27(+)	11.08(+)	9.88(+)	35.63(+)	30.24(+)
Waveform21	14.64	15.41	**14.36**	17.14(+)	15.62(+)	15.88(+)	15.61(+)	15.65(+)	15.62	15.51	24.45(+)
Waveform40	15.28	15.58	15.18	17.29(+)	15.26	15.46	15.26	**15.16**	15.49	16.63	25.40(+)
Yeast	39.50	40.04	**39.49**	40.07	42.51(+)	43.67(+)	41.33	42.78(+)	40.61	54.76(+)	47.25(+)
Win/Draw/Loss		6/15/0	6/14/1	13/8/0	7/13/1	7/13/1	6/14/1	7/13/1	7/14/0	12/9/0	18/3/0

'+/-' shows that the performance of RPRDE is statistically better/worse (by using the statistical test proposed in [36]) than that algorithm for that dataset. RPRD ensembles generally perform similar to or better than other ensemble methods. P represents that the base classifier is unpruned J48, whereas U represents that the base classifier is unpruned J48.

TABLE 6
Average Classification Errors (in %) for Different Ensembles Methods on Different Datasets, the Ensemble Size 10, Class Noise 10%, Bold Numbers Show Best Performance

Data	RPRDE	RD ensembles	RP ensembles	Bagging	AdaBoost (U)	AdaBoost (P)	Multi-boosting(U)	Multi-boosting(P)	Random Forests	RDT	Single tree
Balance	**15.49**	15.34	16.81	19.74(+)	26.07(+)	24.99(+)	21.82(+)	22.37(+)	21.48(+)	30.52(+)	25.14(+)
Breast Cancer	5.67	5.47	**3.70**(-)	6.72	8.75	7.29	6.35	5.85	7.63	11.68(+)	8.49(+)
Ecoli	17.69	17.40	**17.10**	19.70	25.27(+)	22.46(+)	21.42(+)	21.08(+)	21.18(+)	61.41(+)	24.79(+)
Glass	31.94	**31.39**	33.24	34.63	37.13	36.12	37.13	35.27	31.94	40.52(+)	40.93(+)
Ionosphere	10.85	10.71	**10.68**	11.59	14.09(+)	15.54(+)	13.58(+)	13.79(+)	11.19	21.18(+)	16.42(+)
Letter	**7.81**	8.63(+)	14.94(+)	11.15(+)	12.92(+)	12.29(+)	11.41(+)	10.04(+)	11.81(+)	42.33(+)	18.41(+)
Optical	**3.94**	4.40(+)	6.06(+)	6.70(+)	6.01(+)	6.24(+)	6.15(+)	5.58(+)	5.35(+)	14.71(+)	18.43(+)
Pendigits	**1.29**	1.42	1.84(+)	3.42(+)	3.46(+)	3.67(+)	3.06(+)	2.95(+)	2.61(+)	16.41(+)	11.42(+)
Phoneme	**13.49**	15.22(+)	16.89(+)	14.71(+)	18.44(+)	17.87(+)	15.24(+)	15.87(+)	15.09(+)	31.26(+)	18.80(+)
Pima-diabetes	28.44	27.69	**25.71**	27.32	29.50	30.28	28.04	29.36	28.29	32.81	29.66
Ring-Norm	5.06	6.48(+)	**3.85**(-)	6.43(+)	8.57(+)	8.77(+)	6.72(+)	6.69(+)	7.08(+)	16.71(+)	12.29(+)
Satimage	**10.80**	11.24	11.27	12.35(+)	12.77(+)	12.54(+)	12.17(+)	11.66(+)	11.47(+)	31.16(+)	22.98(+)
Segment	**3.36**	3.54	4.58(+)	5.20(+)	7.02(+)	6.63(+)	5.61(+)	5.35(+)	5.66(+)	17.73(+)	10.93(+)
Sonar	**27.59**	27.02	28.45	29.81	30.96	28.08	30.96	28.67	24.42	32.31	34.61(+)
Spambase	9.62	8.63	11.68(+)	**8.36**(-)	11.38(+)	11.28(+)	8.96	8.74	8.80	14.61(+)	11.63(+)
Two-Norm	5.36	7.21(+)	**5.21**	7.36(+)	10.32(+)	10.11(+)	7.89(+)	8.29(+)	8.12(+)	12.91(+)	17.88(+)
Vehicle	28.04	**27.71**	32.85	28.42	29.69	31.13	28.02	29.73	29.33	36.11(+)	34.34
Vowel	**15.40**	18.31(+)	18.71(+)	23.30(+)	22.52(+)	22.72(+)	23.06(+)	23.21(+)	22.92(+)	49.95(+)	33.49(+)
Waveform21	**18.11**	18.70	18.18	19.78(+)	21.08(+)	20.97(+)	19.78(+)	20.49(+)	19.84(+)	25.18(+)	29.45(+)
Waveform40	**19.05**	19.13	21.60(+)	21.54(+)	20.08(+)	20.49(+)	21.22(+)	20.30(+)	20.49(+)	28.86(+)	31.68(+)
Yeast	42.90	**42.56**	44.51	43.67	47.83(+)	47.78(+)	45.35(+)	44.56	45.16(+)	58.61(+)	52.17(+)
Win/Draw/Loss		6/15/0	8/11/2	12/8/1	16/5/0	16/5/0	15/6/0	14/7/0	14/7/0	19/2/0	19/2/0

'+/-' shows that the performance of RPRDE is statistically better/worse (by using the statistical test proposed in [36]) than that algorithm for that dataset. RPRD ensembles generally perform similar to or better than other ensemble methods. P represents that the base classifier is unpruned J48, whereas U represents that the base classifier is unpruned J48.

TABLE 7
Classification Errors (in %) for Different Ensemble Methods on Different Datasets, Bold Numbers Show Best Performance, the Ensemble Size 100, the Class Noise is 10%

Data	RPRDE	RD ensembles	RP ensembles	Bagging	AdaBoost (U)	AdaBoost (P)	Multi-boosting(U)	Multi-boosting(P)	Random Forests	RDT	Single tree
Balance	13.61	13.86	**12.62**	18.85(+)	26.35(+)	25.76(+)	23.41(+)	22.37(+)	21.72(+)	24.83(+)	25.14(+)
Breast Cancer	4.98	5.32	**3.50**	5.90	8.69(+)	7.71(+)	8.57(+)	7.82(+)	6.18	11.87(+)	8.49(+)
Ecoli	15.44	**15.62**	14.67	18.22	22.89(+)	22.45(+)	21.48(+)	20.58(+)	18.58	58.82(+)	24.79(+)
Glass	29.63	28.61	**29.26**	31.94	34.72	32.54	33.89	30.56	30.09	36.56(+)	40.93(+)
Ionosphere	8.92	9.82	**7.72**	10.68	11.76	11.07	11.59	11.09	9.54	19.31(+)	16.42(+)
Letter	**4.93**	5.67(+)	7.91(+)	8.63(+)	11.52(+)	11.69(+)	8.51(+)	7.99(+)	7.74(+)	19.61(+)	18.41(+)
Optical	2.58	2.82(+)	**2.15**	4.66(+)	2.27	2.46	2.45	2.48	2.22	10.91(+)	18.43(+)
Pendigits	**0.71**	1.03(+)	1.01(+)	2.48(+)	1.24(+)	1.47 (+)	1.20(+)	1.44(+)	1.21(+)	13.91(+)	11.41(+)
Phoneme	**12.38**	14.69(+)	16.11(+)	13.85(+)	18.44(+)	17.92(+)	18.44(+)	17.89(+)	13.28(+)	30.51(+)	18.80(+)
Pima-diabetes	25.63	26.08	**25.29**	25.77	29.56(+)	29.86(+)	29.40(+)	30.02(+)	26.36	31.14(+)	29.66
Ring-Norm	1.98	2.77(+)	**1.62**(-)	4.70(+)	3.00(+)	3.23(+)	2.74(+)	2.94(+)	2.99(+)	12.11(+)	12.29(+)
Satimage	9.36	9.69	10.15	10.84(+)	9.40	9.34	9.54	9.40	**9.33**	30.42(+)	22.99(+)
Segment	**2.73**	2.84	3.62(+)	4.29(+)	6.20(+)	5.95(+)	4.52(+)	4.15(+)	4.20(+)	15.82(+)	10.94(+)
Sonar	25.96	26.06	25.19	27.51	26.92	22.32	27.12	22.99	**22.12**	29.38	34.61(+)
Spambase	7.98	7.76	10.28(+)	7.73	10.48(+)	11.25(+)	9.15(+)	9.06(+)	**7.04**	13.32(+)	11.63(+)
Two-Norm	2.74	3.53(+)	**2.52**	3.82(+)	3.90(+)	4.10(+)	3.59(+)	3.68(+)	3.62(+)	10.91(+)	17.88(+)
Vehicle	25.97	26.56	30.28	27.24	**26.70**	27.07	26.60	26.87	27.33	32.67(+)	34.34(+)
Vowel	11.02	12.33	**10.87**	20.06(+)	18.29(+)	16.55(+)	17.86(+)	16.38(+)	15.46(+)	37.62(+)	33.49(+)
Waveform21	15.07	15.89	**14.57**	17.15(+)	16.23(+)	16.12(+)	15.70	15.72	16.04(+)	22.78(+)	29.45
Waveform40	15.51	15.75	**15.49**	17.46(+)	15.83	15.98	15.46	15.58	15.42	21.38(+)	31.68(+)
Yeast	40.64	40.66	**40.01**	42.25	45.84(+)	45.26(+)	43.97	42.99	42.39	57.61(+)	52.17(+)
Win/Draw/Loss		6/15/0	5/15/1	12/9/0	14/7/0	14/7/0	12/9/0	12/9/0	9/12/0	20/1/0	20/1/0

'+/-' shows that the performance of RPRDE is statistically better/worse (by using the statistical test proposed in [36]) than that algorithm for that dataset. RPRD ensembles generally perform similar to or better than other ensemble methods, however, their competitive advantage is more for smaller ensembles. P represents that the base classifier is unpruned J48, whereas U represents that the base classifier is unpruned J48.

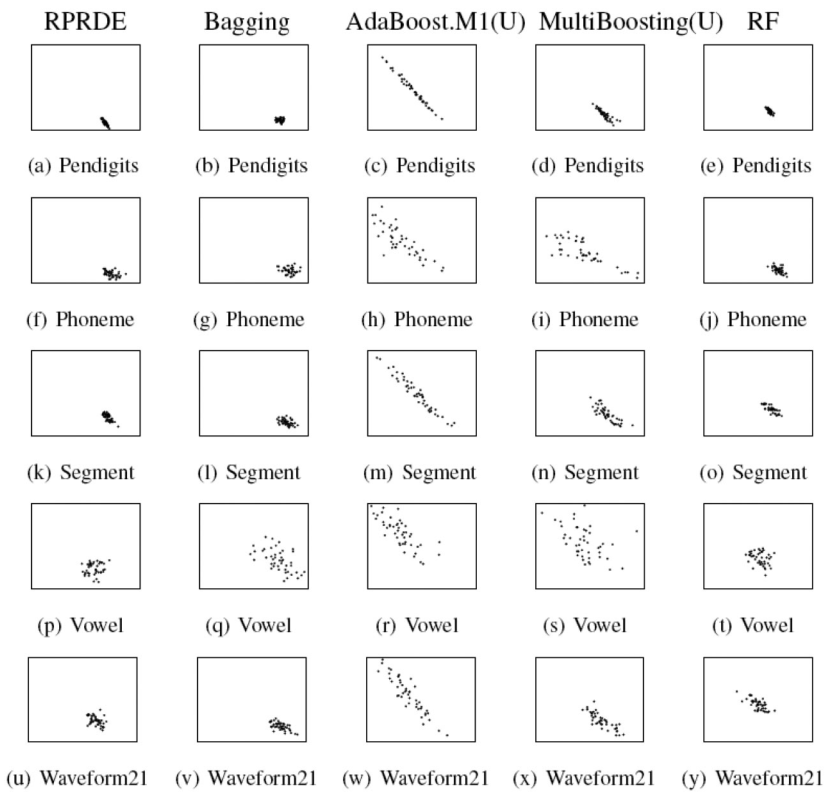

Fig. 6. Kappa-error plots for four ensemble methods, First column - RPRDE, second column - Bagging, third column - AdaBoost.M1, fourth column - Multiboostinging and last column RF. x-axis - Kappa (κ), y-axis - the average error of the pair of classifiers. Axes scales are constant for various ensemble methods for a particular dataset (each row). Lower κ represents a higher diversity. The plots suggest that RPRDE classifiers are accurate with reasonable diversity. U represents that the base classifier is unpruned J48.

as AdaBoost.M1(U), MutiBoosting(U) and Random Forests, however, generally RPRDE is more diverse than Bagging. Classifiers created by using RPRDE and Bagging generally have similar accuracy performance, whereas they are generally more accurate than all other ensemble methods. One may conclude that RPRDE behaviour is midway between these two types of methods, Bagging (classifiers; more accurate, less diverse) and Adaboost.M1 (classifiers; less accurate, more diverse). RPRDE is able to improve diversity, but to a lesser degree than Adaboost.M1 and Random Forests, without affecting accuracy of individual classifiers as much as Adaboost.M1, Random Forests and MutiBoosting. Kappa-error plots indicate that *accurate classifiers with reasonable diversity* is the reason for the success of RPRDE.

6.6 The Study of Tree Growing Phase Time

We carried out experiments to study the tree growing phase for different ensemble methods. Experiments were carried out on five datasets (Pen, Phoneme, Segment, Vowel and Waveform21). For each dataset, in each run of 5×2 cross-validation, 100 trees were created. In other words, 1000 trees were created for each dataset. The average tree growing phase times for different ensemble methods for different datasets are presented in Table 8. As expected the tree growing phase time for RPRDE is the highest. As discussed in subsection 5.2, RPRDE is using new RP features, the creation of these new features adds extra computational cost. These new features are added with the discretized original features. Hence, the feature size of the datasets in RPRDE is

TABLE 8
The Average Tree Growing Time in Sec., for Different Ensemble Methods

Data	RPRDE	Bagging	AdaBoost (U)	AdaBoost (P)	Multi-boosting(U)	Multi-boosting(P)	Random Forests
Pen	1.82	0.97	1.27	1.14	1.31	1.18	0.31
phoneme	0.56	0.24	0.42	0.34	0.41	0.34	0.12
Segment	0.49	0.21	0.29	0.27	0.28	0.26	0.08
Vowel	0.24	0.12	0.20	0.18	0.21	0.19	0.05
Waveform21	2.31	1.52	2.18	2.10	2.12	2.04	0.45

P represents that the base classifier is unpruned J48, whereas U represents that the base classifier is unpruned J48.

TABLE 9
Number of RP Attributes at Top Levels of RPRD Trees

Data	In top three levels	In top five levels
Pen	1	2
phoneme	1	1
Segment	2	3
Vowel	1	2
Waveform21	0	1

more, this also contributes to the high tree growing phase time.

6.7 The Experimental Study of the Structure of RPRD Trees

We discussed in subsection 5.1 that the success of RPRDE is due to the fact that PRRD trees have good combination of RD attributes and RP attributes. In other words, some of the RP features are selected at high levels of RPRD trees. We carried out experiments to study the structure of RPRD trees. We studied how many RP features were selected at high levels (top three and top five) of RPRD trees. Experiments were carried out on five datasets (Pen, Phoneme, Segment, Vowel and Waveform21). For each dataset, in each run of 5×2 cross-validation, 100 trees were created. In other words, 1000 trees were created for each dataset. The average results are presented in Table 9. Results suggest that except Waveform21, all the decision trees for all the other datasets have at least one RP attribute in top three levels. For all datasets, decision trees have atleast one RP attribue in top five levels. For three datasets (Pendigits, Segment and Vowel), decision trees have at least two attributes in top five levels. This suggests that generally RPRD trees have a good combination of RP attributes and RP attributes that helps in creating diverse trees.

7 Conclusion and Future Work

In the proposed work, we developed RPRDE method that creates an ensemble of linear multivariate decision trees by using a univariate decision tree algorithm. We propose the Random Disrectization (RD) process which creates random discretized features. We showed that ensembles created by RD trees have good representational power. We combined two data transformation processes, Random Projection (RP) and Random Discretization (RD), to create RPRD ensembles. RPRD trees are linear multivariate decision trees; hence, we expect that RPRD ensembles consisting of RPRD trees have good representational power. Experimental results on the synthetic dataset with a diagonal concept prove this point. The comparative study against the other popular ensemble methods shows the superiority of RPRDE. Results suggest that RPRDE is more useful for small ensembles. Experiments also suggest that RPRDE is robust to the noisy data. Ensembles of omnivariate decision trees (using univariate decision tree algorithms) will be an interesting research field as that will have more representational power (omnivariate decision trees can have nonlinear decision surfaces). In the proposed ensemble method, new diverse datasets are created, other classifiers instead of decision trees can be used on these datasets. The study of various classifiers with these diverse datasets will be one of the future directions of this work.

References

[1] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. 1st Int. Workshop MCS*, vol. 1857, Cagliari, Italy, 2000, pp. 1–15.

[2] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.

[3] K. Tumer and J. Ghosh, "Error correlation and error reduction in ensemble classifiers," *Connect. Sci.*, vol. 8, no. 3, pp. 385–404, 1996.

[4] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[5] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[6] G. I. Webb, "Multiboosting: A technique for combining boosting and wagging," *Mach. Learn.*, vol. 40, no. 2, pp. 159–196, 2000.

[7] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[8] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[9] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, Oct. 2006.

[10] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Mach. Learn.*, vol. 40, no. 2, pp. 1–22, 2000.

[11] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

[12] W. Fan, H. Wang, P. S. Yu, and S. Ma, "Is random model better? On its accuracy and efficiency," in *Proc. 3rd IEEE Int. Conf. Data Mining*, Hawthorne, NY, USA, 2003, pp. 51–58.

[13] W. Fan, J. McCloskey, and P. S. Yu, "A general framework for accurate and fast regression by data summarization in random decision trees," in *Proc. 12th ACM SIGKDD Int. Conf. KDD*, Philadelphia, PA, USA, 2006, pp. 136–146.

[14] R. E. Banfield *et al.*, "A comparison of ensemble creation techniques," in *Proc. 5th Int. Conf. MCS*, Cagliari, Italy, 2004.

[15] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," in *Proc. 14th Int. Conf. Mach. Learn.*, Burgos, Spain, 1997.

[16] W. Johnson and J. Lindenstrauss, "Extensions of Lipshitz mapping into Hilbert space," in *Proc. Conf. Modern Anal. Probab.*, New Haven, CT, USA, 1984, pp. 189–206.

[17] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson-Lindenstrauss lemma," Int. Comput. Sci. Institute, Berkeley, CA, USA, Tech. Rep. TR-99-006, 1999.

[18] J. R. Quinlan, *C4.5: Programs for Machine Learning*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[19] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Belmont, CA, USA: Wadsworth International Group, 1984.

[20] C. E. Brodley and P. E. Utgoff, "Multivariate decision trees," *Mach. Learn.*, vol. 19, no. 1, pp. 45–77, Apr. 1995.

[21] V. S. Iyengar, "HOT: Heuristics for oblique trees," in *11th Int. Conf. Tools Artif. Intell.*, IEEE Press, Chicago, IL, USA, 1999, pp. 91–98.

[22] D. G. Heath, S. Kasif, and S. Salzberg, "Induction of oblique decision trees," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 1993, pp. 1002–1007.

[23] S. K. Murthy, S. Kasif, S. Salzberg, and R. Beigel, "Oc1: A randomized induction of oblique decision trees," in *Proc. 11th Nat. Conf. Artif. Intell.*, 1993, pp. 322–327.

[24] R. Arriaga and S. Vempala, "An algorithmic theory of learning: Robust concepts and random projection," *Mach. Learn.*, vol. 63, no. 2, pp. 161–182, 2006.

[25] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proc. 20th Int. Conf. Mach. Learn.*, Washington, DC, USA, 2003, pp. 186–193.

[26] A. Schclar and L. Rokach, "Random projection ensemble classifiers," in *Proc. 11th Int. Conf. Enterprise Inform. Syst.*, Milan, Italy, 2009, pp. 309–316.

[27] Y. Freud and R. E. Schpire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, 1996.

[28] A. Ahmad, S. M. Halawani, and I. Albidewi, "Consistency of randomized and finite sized decision tree ensembles," *Pattern Anal. Appl.*, vol. 17, no. 1, pp. 97–104, Feb. 2014.

[29] G. Biau and L. Devroye, "Consistency of random forests and other averaging classifiers," *J. Mach. Learn. Res.*, vol. 9, pp. 2015–2033, Sept. 2008.

[30] D. Fradkin and D. Madigan, "Experiments with random projections for machine learning," in *Proc. 9th ACM SIGKDD Int. Conf. KDD*, New York, NY, USA, 2003, pp. 517–522.

[31] M. F. Balcan, A. Blum, and S. Vempala, "Kernels as features: On kernels, margins, and low-dimensional mappings," *Mach. Learn.*, vol. 65, no. 1, pp. 79–94, 2006.

[32] M. F. Balcan and A. Blum, "On a theory of learning with similarity functions," in *Proc. 23rd Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, 2006.

[33] M. R. Rwebangira, "Techniques for exploiting unlabeled data," Ph.D. dissertation, School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2008.

[34] M. Hall *et al.*, "The WEKA data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[35] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no 1, pp. 1895–1923, Oct. 1998.

[36] E. Alpaydin, "Combined 5 x 2 cv f test comparing supervised classification learning algorithms," *Neural Comput.*, vol. 11, no. 8, pp. 1885–1892, 1999.

[37] S. Dasgupta, "Learning mixtures of Gaussians," in *Proc. 40th Annu. IEEE Symp. Found. Comput. Sci.*, New York, NY, USA, 1999, pp. 634–644.

[38] D. Achlioptas, "Database-friendly random projections," in *Proc. ACM Symp. Principles of Database Systems*, 2001, pp. 274–281.

[39] A. Frank and A. Asuncion. (2010). *UCI machine learning repository* [Online]. http://archive.ics.uci.edu/ml

[40] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," in *Proc. 14th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 1997, pp. 211–218.

Amir Ahmad received the Ph.D. degree in computer science from the University of Manchester. He is currently with King Abdulaziz University, Rabigh, Saudi Arabia, as an Assistant Professor. His current research interests include data mining and nanotechnology.

Gavin Brown is a Lecturer in Machine Learning at the University of Manchester. His research has gravitated around information fusion mechanisms—including ensemble learning, feature selection and extraction using information theoretic and probabilistic approaches. His current research interests include biohealth informatics and adaptive compiler technologies.