

A Replicator Dynamics Approach to Collective Feature Engineering in Random Forests

Khaled Fawgreh, Mohamed Medhat Gaber and Eyad Elyan

Abstract It has been demonstrated how random subspaces can be used to create a Diversified Random Forest, which in turn can lead to better performance in terms of predictive accuracy. Motivated by the fact that each subforest is built using a set of features that can overlap with those sets of features in other subforests, we hypothesise that using *Replicator Dynamics* can perform a collective feature engineering, by allowing subforests with better performance to grow and those with lower performance to shrink. In this paper, we propose a new method to further improve the performance of Diversified Random Forest using Replicator Dynamics which has been used extensively in evolutionary game dynamics. A thorough experimental study on 15 real datasets showed favourable results, demonstrating the potential of the proposed method. Some experiments reported a boost in predictive accuracy of over 10 % consistently, evidencing the effectiveness of the iterative feature engineering achieved through the *Replicator Dynamics* procedure.

1 Introduction

Diversified Random Forest (DRF) [8] is an extension of Random Forest (RF) which is an ensemble learning technique used for classification and regression. Ensemble learning is a supervised machine learning paradigm where multiple models are used to solve the same problem [23]. Since single classifier systems have limited predictive performance [20, 23, 25, 29], ensemble classification was developed to overcome this limitation [20, 23, 25], and thus boosting the accuracy of classification. In such an ensemble, multiple classifiers are used. In its basic mechanism, majority

K. Fawgreh · M.M. Gaber (✉) · E. Elyan
School of Computing Science and Digital Media, Robert Gordon University,
Garthdee Road, Aberdeen, AB10 7GJ, UK
e-mail: m.gaber1@rgu.ac.uk

K. Fawgreh
e-mail: k.fawagreh@rgu.ac.uk

E. Elyan
e-mail: e.elyan@rgu.ac.uk

voting is then used to determine the class label for unlabelled instances where each classifier in the ensemble is asked to predict the class label of the instance being considered. Once all the classifiers have been queried, the class that receives the greatest number of votes is returned as the final decision of the ensemble.

Boosting, bagging, and stacking are the three widely used and adopted ensemble methods. Boosting is an incremental process of building a sequence of classifiers, where each classifier has the incorrectly classified instances of the previous one in the sequence emphasised. AdaBoost [11] is the representative of this class of techniques. However, AdaBoost is prone to overfitting, due to the nature of the process. The other class of ensemble approaches is the bootstrap aggregating (bagging) [4]. Bagging involves building each classifier in the ensemble using a randomly drawn bootstrap sample of the data, having each classifier giving an equal vote when classifying unlabelled instances. Bagging is known to be more robust than boosting against model overfitting. Random Forest (RF) is the main representative of bagging [5]. Stacking (sometimes called stacked generalisation) extends the cross-validation technique that partitions the dataset into a held-in data set and a held-out data set; training the models on the held-in data; and then choosing whichever of those trained models performs best on the held-out data. Instead of choosing among the models, stacking combines them, thereby typically getting performance better than any single one of the trained models [28]. It is worth noting that each of these three methods diversify among the classifiers of the ensemble. Diversity plays an important role in the success of the ensemble.

Since RF has been proved to be the state-of-the-art ensemble classification technique, and since it has been proven empirically that ensembles tend to yield better results when there is a significant diversity among the models [1, 6, 18, 26], DRF was developed as an extension of RF by injecting a new level of diversity [8]. An observation in DRF has motivated the work proposed in this paper; that subforests created in DRF can exhibit varying discriminative power according to the randomly drawn subspace (more details about DRF is given later in the paper). We hypothesise that if subforests with relatively better accuracy are allowed to grow, and those with less accuracy can shrink, we can perform inherit feature engineering of the ensemble in a collective manner. This is done by means of subforest growth and shrinking that can be looked at as emphasising a set of features collectively over other sets. This unique way of feature engineering allows for feature interactivity to take place, thus providing a new method that favours feature co-existence in some subforests. As such, this paper investigates how to further boost the performance of DRF by using *Replicator Dynamics* that provides the mechanism for growing and shrinking subforests.

This paper is organised as follows. First, an overview of DRF is presented in Sect. 2. This is followed by Sect. 3 that presents a brief introduction to Replicator Dynamics. Section 4 demonstrates how Replicator Dynamics can be utilised to boost the performance of DRF. Experimental results demonstrating the superiority of the proposed method is detailed in Sect. 5. In Sect. 6, we describe related work. The paper is then concluded with a summary and pointers to future directions in Sect. 7.

2 Diversified Random Forests: An Overview

As detailed in [8], creating an DRF proceeds as follows. First, from the training set, we create a number of random subspaces. The number of subspaces is determined by the following equation:

$$Subspaces = \alpha \times Z \quad (1)$$

where α denotes the subspace factor such that $0 < \alpha \leq 1$, and Z is the size of DRF to be created. Each subspace will contain a fixed randomized subset of the total number of features and will correspond to a sub-forest. A projected training dataset will be then created for each subspace and will be used to create the trees in the corresponding sub-forest. The following equation calculates the number of trees in each sub-forest:

$$Trees = \frac{Z}{Subspaces} \quad (2)$$

Next, a weight is assigned to each projected training dataset using the *Absolute Predictive Power (APP)* given by Cuzzocrea et al. [7]. Given a dataset S , the *APP* is defined by the following equation:

$$APP(S) = \frac{1}{|Att(S)|} \times \sum_{A \in Att(S)} \frac{I(S, A)}{E(S)} \quad (3)$$

where $E(S)$ is the entropy of a given dataset S having K instances and $I(S, A)$ is the information gain of a given attribute A in a dataset S . $E(S)$ is a measure of the uncertainty in a random variable and is given by the following equation:

$$E(S) = \sum_{i=1}^K -p_i(x_i) \log_2 p_i(x_i) \quad (4)$$

where x_i refers to a generic instance of S and $p_i(x_i)$ denotes the probability that the instance x_i occurs in S . $I(S, A)$ is given by the following equation:

$$I(S, A) = E(S) - \sum_{v \in Val(A)} \left(\frac{|S_v|}{|S|} \right) E(S_v) \quad (5)$$

where $E(S)$ denotes the entropy of S , $Val(A)$ denotes the set of possible values for A , S_v refers to the subset of S for which A has the value v , and $E(S_v)$ denotes the entropy of S_v .

The weight given in Eq. 3 above will be inherited by the corresponding sub-forest and will be used in the voting process. This means that the standard voting technique, currently used in the standard RF, is going to be replaced by a weighted voting technique, in order to classify the instances in the testing dataset.

3 Replicator Dynamics

Replicator Dynamics (RD) is a deterministic monotone non-linear and also non-innovative game dynamic (older solutions are obsolete) used in evolutionary game theory [16, 27]. It provides a convenient way to represent selection among a population of diverse types. To illustrate how it works, assume that selection occurs between periods after dividing time into discrete periods. The proportion of each type in the next period is given by the replicator equation as a function of the type's payoffs and its current proportion in the population. Types that score above the average payoffs increase in proportion, while types that score below the average payoffs decrease in proportion. The amount of increase or decrease depends on a type's proportion in the current population and on its relative payoffs.

The most general continuous form is given by the differential equation

$$\dot{x}_i = x_i[f_i(x) - \phi(x)] \quad (6)$$

such that

$$\phi(x) = \sum_{j=1}^n x_j f_j(x) \quad (7)$$

where x_i is the proportion of type i in the population, $x = (x_1, \dots, x_n)$ is the vector of the distribution of types in the population, $f_i(x)$ is the fitness of type i (which is dependent on the population), and $\phi(x)$ is the average population fitness (given by the weighted average of the fitness of the n types in the population).

In the next section, we shall see how to apply these equations in order to boost the performance of an DRF. To the best of our knowledge, RD has never been used before in ensemble learning.

4 Applying Replicator Dynamics to an DRF

As was detailed in Sect. 2, the population is n sub-forests where each sub-forest represents a type. The discrete periods mentioned in the previous section correspond to loop iterations. At each iteration, the accuracy of the sub-forest being processed is compared with the average accuracy of the entire DRF which is calculated as the average accuracy of the sub-forests. If it is greater, then the size of the sub-forest grows and if it less, the size shrinks.

For growing and shrinking the sub-forest, two variations will be used. In the first one, the size grows/shrinks by a fixed number as shown in the following equations:

$$\text{treesToAdd} = \beta \quad (8)$$

$$\text{treesToRemove} = \gamma \quad (9)$$

In the second variation, the sub-forest grows/shrinks by adding/removing a variable number of trees according to the following equations:

$$treesToAdd = \lfloor ((subforestAccuracy(i) - DRFAccuracy) \times numTrees) \rfloor \quad (10)$$

$$treesToRemove = \lfloor ((DRFAccuracy - subforestAccuracy(i)) \times numTrees) \rfloor \quad (11)$$

where $subforestAccuracy(i)$ refers to the accuracy of subforest(i) being processed, and $numTrees$ refers to the initial number of trees that was used to construct the sub-forest (given by Eq. 2 above). The $floor$ function maps a real number to the largest previous integer. The $DRFAccuracy$ refers to the average accuracy of the entire DRF which can be calculated as follows:

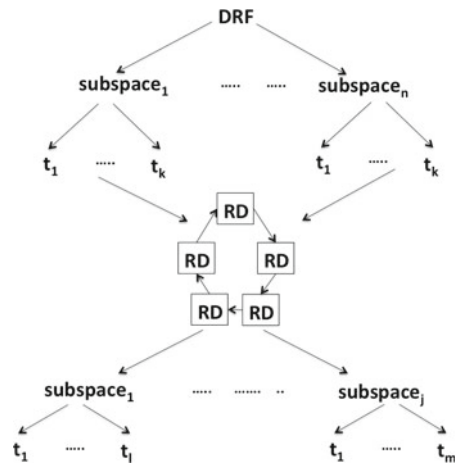
$$DRFAccuracy = \frac{1}{Subspaces} \sum_{i=1}^{Subspaces} subforestAccuracy(i) \quad (12)$$

where the constant $Subspaces$ is given by Eq. 1.

Figure 1 outlines the main steps involved in applying RD to a DRF to further optimise its performance. As shown in the figure, RD is performed over DRF in an iterative procedure. Thus, we can identify two main stages to the method proposed in this paper: (1) building a DRF; and (2) applying RD a preset number of iterations. The outcome is a number of subforests, each with potentially a different number of trees than the number initially set when building the DRF in the first step.

Algorithm 1 illustrates how RD can be applied to a DRF to grow or shrink the subs-forests. At each iteration, the average accuracy of all subforests is calculated as well as the accuracy of each subforest. This process is used to determine those subforests that have an above average accuracy to grow them further. On the other hand, subforests that have accuracy below the calculated average are forced to shrink.

Fig. 1 Applying RD to an DRF



The process performs inherit collective feature engineering, as subforests that grow have the features that collectively produce accurate trees.

Algorithm 1 Applying Replicator Dynamics to a DRF

```

{User Settings}
input DRF
input numberOfIterations
{Process}
for  $i = 1 \rightarrow \text{numberOfIterations}$  do
  for  $j = 1 \rightarrow \text{DRF.numberOfSubforests}()$  do
     $\text{DRFAccuracy} \leftarrow \text{CalculateDRFAccuracy}(\text{DRF})$ 
     $\text{subforestAccuracy} \leftarrow \text{CalculateSubforestAccuracy}(\text{DRF.subforest}(j))$ 
    if ( $\text{subforestAccuracy} > \text{DRFAccuracy}$ ) then
      Apply Eq. 8 or 10 to determine treestoAdd
       $\text{AddTrees}(\text{DRF.subforest}(j), \text{treestoAdd})$ 
    else if ( $\text{subforestAccuracy} < \text{DRFAccuracy}$ ) then
      Apply Eq. 9 or 11 to determine treesToRemove
       $\text{RemoveTrees}(\text{DRF.subforest}(j), \text{treestoRemove})$ 
    end if
  end for
end for
{Output}
A new DRF with grown or shrunk sub-forests
  
```

5 Experimental Study

Several experiments were performed on 15 real datasets from the UCI repository [2]. All the DRFs created had an initial size of 500 trees.

We used 2 subspace factors of 2 and 4 %. According to Eq. 1, these factors produced DRFs with 10 and 20 sub-forests respectively. We used a random 70 % of the features for each subspace, which has proved empirically to lead good performance in the traditional version of DRF. By Eq. 2, each sub-forest contained 50 trees for the DRF with 10 sub-forests, and 25 trees for the DRF with 20 sub-forests. For the number of iterations (refer to *Number of Iterations* in Algorithm 1 above), we used 25, 50, 100, 150, and 1000 iterations. Table 1 lists the different experimental scenarios that will be discussed in details later on in the paper. It is worth pointing out that results reported in this section captured all the experiments that have been carried out. However, due to space limitation, the results of each scenario are summarized in a separate table, followed by a detailed table of winning solutions that represent the significant difference in accuracy that has been achieved frequently.

In the first set of experiments, we used the first variation given by Eqs. 8 and 9 above. Here we chose $\beta = \gamma = 1$, therefore, only 1 tree may be added/removed in each iteration. A new experiment was performed on each of the 10 scenarios listed

Table 1 Scenario reference table

Scenario#	Number of sub-forests	Number of trees per sub-forest	Number of iterations
1	10	50	25
2	10	50	50
3	10	50	100
4	10	50	150
5	10	50	1000
6	20	25	25
7	20	25	50
8	20	25	100
9	20	25	150
10	20	25	1000

in Table 1. We then compared the performance of DRF before and after applying RD to see if there is a performance gain and recorded the number of wins, losses, and ties sorted by number of wins in descending order as shown in Table 2. As shown in the table, the number of wins of the new DRF (after applying RD) exceeded the number of losses in the majority of the scenarios. By wins we refer to the number of datasets where the new DRF outperformed the initial DRF. In fact, with the exception of scenario 2, the adoption of RD has consistently shown superiority. It is worth noting that the number of iterations in this scenario is only 50. This suggests that the larger number of iterations can result in a better performance. We argue that with the larger number of iterations, the feature engineering process is stabilised.

Table 2 First experiments set results

Scenario	Wins	Losses	Ties
8	6	4	5
9	6	4	5
4	6	2	7
3	6	3	6
6	6	3	6
1	5	1	9
7	5	4	6
5	5	4	6
10	4	4	7
2	3	4	8

Accuracy of the winning datasets is shown in Table 3. As demonstrated in the table, the improvement in the accuracy after applying RD can be very significant (e.g. in scenario 5, for the squash-unstored dataset, the accuracy was boosted from 61.1 to 77.7%). This provides the evidence of the positive effect of the feature engineering provided by the RD process.

In the second set of experiments and using again the first variation, performance was compared with traditional RFs of identical size (500 trees). Table 4 shows the results. Once again, the number of wins exceeded the number of losses in the majority of the scenarios. In this experiment, scenario 10 that was experimented with 1000 iterations of RD has shown the best performance in terms of predictive accuracy. This also confirms that feature engineering through RD requires a large number of iterations to stabilise, as we argued in the first set of experiments.

Table 5 shows the accuracy of the winning datasets. As DRF has already improved the performance of the ensemble, the results in this table show more datasets that have had their accuracy boosted. It is important to note that the boost in the accuracy is due to two factors: (1) the diversification achieved through random subsampling; and (2) the feature engineering provided through RD.

The third and fourth set of experiments were performed using the second variation given by Eqs. 10 and 11 above where a fraction of the initial number of trees in the sub-forest may be added or removed. Similar to Table 2, Table 6 compares the performance of DRF with itself before and after applying RD. This represents the typical implementation of the RD, where the subforest with a considerable over average accuracy can grow with more than one tree in each iteration.

Table 7 shows the accuracy of the winning datasets. As demonstrated in the previously discussed experiments, the increase in accuracy is significant in some of the datasets, suggesting the effectiveness of RD in the feature engineering process.

Likewise, Table 8 is similar to Table 4 where the DRFs (after applying RD) are compared with traditional RFs of identical size. As was the case with the first and second set of experiments, favourable results were achieved since, as demonstrated in these tables, the number of wins exceeded the number of losses in the majority of the scenarios. We also note that the two scenarios with the highest number of iterations (1000 iterations) have demonstrated the best performance. This confirms that a large number of iterations is required for the feature engineering to be effective, as argued earlier in this experimental study.

Accuracy of the winning datasets is shown in Table 9. Significant boost in the performance has been demonstrated as shown in the table.

The extensive experimental study discussed in this section has demonstrated the effectiveness of the *Replicator Dynamics* in collective feature engineering. As subforests that contain subspaces with high discriminative power features can grow—so that their contributions to the vote also grow. On the other hand, subforests that contain subspaces with low discriminative power features can shrink, and potentially disappear.

Table 3 First experiments set winning datasets accuracy

Scenario	Dataset	Accuracy before (%)	Accuracy after (%)
8	Squash-unstored	61.11	77.78
	Vote	96.62	97.30
	Vehicle	72.92	74.65
	Eucalyptus	23.60	29.60
	Audit	96.76	96.91
	Soybean	77.59	78.88
9	Squash-unstored	61.11	77.78
	Vehicle	72.57	73.96
	Eucalyptus	280	28.80
	Sonar	00.00	01.41
	Audit	96.62	97.06
	Soybean	73.71	85.78
4	Squash-unstored	61.11	72.22
	Squash-stored	44.44	50.00
	White clover	61.90	66.67
	Vehicle	72.57	72.92
	Eucalyptus	20.80	26.40
	Credit	77.94	78.82
3	Squash-unstored	61.11	66.67
	Glass	10.96	12.33
	Diabetes	74.33	74.71
	Vehicle	70.83	72.57
	Eucalyptus	18.00	24.00
	Sonar	00.00	01.41
6	Squash-unstored	61.11	66.67
	Diabetes	73.56	75.10
	Vehicle	72.22	73.26
	Eucalyptus	22.00	26.40
	Credit	75.29	76.18
	Audit	96.47	96.76
1	Diabetes	72.41	73.18
	Vehicle	73.61	75.00
	Eucalyptus	21.20	26.40
	Credit	75.29	76.18
	Soybean	73.71	76.29
7	Squash-unstored	61.11	66.66
	Diabetes	73.56	73.95
	Vehicle	73.26	73.61
	Eucalyptus	28.00	29.20
	Audit	97.06	97.21

(continued)

Table 3 (continued)

Scenario	Dataset	Accuracy before (%)	Accuracy after (%)
5	Squash-unstored	50.00	61.11
	White clover	61.90	66.67
	Vehicle	72.92	73.26
	Eucalyptus	17.20	24.00
	Soybean	73.28	75.43
10	Pasture	33.33	41.67
	Squash-unstored	61.11	72.22
	Eucalyptus	23.60	30.40
	Soybean	77.15	79.74
2	Squash-unstored	55.56	61.11
	Eucalyptus	21.60	24.00
	Credit	74.12	76.18

Table 4 Second experiments set results

Scenario	Wins	Losses	Ties
10	9	4	2
1	8	4	3
6	8	4	3
7	8	4	3
3	8	5	2
9	8	4	3
4	7	4	4
2	6	4	5
8	6	6	3
5	5	6	4

6 Related Work

Random Forest has proved to be the state-of-the-art classification technique when compared with other methods in a recent large scale experimental study [10]. Previous work to enhance its performance has been also reported recently in [9]. However, feature engineering through an evolutionary method like *Replicator Dynamics* has not been attempted.

As aforementioned, RD has been used extensively in the domain of evolutionary game theory [13–16, 21, 24, 27]. To a lesser extent, however, it was also used in other domains. In medicine, [19] used RD to analyse fMRI data of the human brain. In mathematical ecology, [3] used RD to describe the interaction of two populations over time. In mathematical biology, [17] used RD in the study of permanence, i.e., the study of the long-term survival of each species in a set of populations. Multi-agent

Table 5 Second experiments set winning datasets accuracy

Scenario	Dataset	Accuracy classic RF (%)	Accuracy DRF (%)
10	Pasture	41.67	41.67
	Squash-unstored	61.11	72.22
	Diabetes	73.56	74.71
	Vehicle	72.92	73.96
	Eucalyptus	21.20	30.40
	Sonar	00.00	01.41
	Credit	75.88	76.76
	Audit	96.32	96.91
	Soybean	75.86	79.74
1	Pasture	41.67	41.67
	Breast-cancer	72.16	75.26
	Vehicle	72.92	75.00
	Eucalyptus	21.20	26.40
	Sonar	00.00	01.41
	Credit	75.88	76.18
	Audit	96.32	96.76
	Soybean	75.86	76.29
6	Pasture	41.67	41.67
	Squash-unstored	61.11	66.67
	Diabetes	73.56	75.09
	Vehicle	72.92	73.26
	Eucalyptus	21.20	26.40
	Credit	75.88	76.18
	Audit	96.32	96.76
	Soybean	75.86	76.29
7	Pasture	41.67	41.67
	Squash-unstored	61.11	66.67
	Breast-cancer	72.16	74.23
	Diabetes	73.56	73.95
	Vehicle	72.92	73.61
	Eucalyptus	21.20	29.20
	Credit	75.88	76.47
	Audit	96.32	97.21
3	Pasture	41.67	41.67
	Squash-unstored	61.11	66.67
	Diabetes	73.56	74.71
	Eucalyptus	21.20	24.00
	Sonar	00.00	01.41
	Credit	75.88	76.76
	Audit	96.32	97.06
	Soybean	75.86	77.15

(continued)

Table 5 (continued)

Scenario	Dataset	Accuracy classic RF (%)	Accuracy DRF (%)
9	Pasture	41.67	41.67
	Squash-unstored	61.11	77.78
	breast-cancer	72.16	74.23
	Vehicle	72.92	73.96
	Eucalyptus	21.20	28.80
	Sonar	00.00	01.41
	Audit	96.32	97.06
	Soybean	75.86	85.78
4	Pasture	41.67	41.67
	Squash-unstored	61.11	72.22
	Breast-cancer	72.16	75.26
	Eucalyptus	21.20	26.40
	Credit	75.88	78.82
	Audit	96.32	96.91
	Soybean	75.86	78.02
2	Pasture	41.67	41.67
	Breast-cancer	72.16	74.23
	Eucalyptus	21.20	24.00
	Credit	75.88	76.18
	Audit	96.32	97.06
	Soybean	75.86	78.02
8	Pasture	41.67	41.67
	Squash-unstored	61.11	77.78
	Vehicle	72.92	74.65
	Eucalyptus	21.20	29.60
	Audit	96.32	96.91
	Soybean	75.86	78.88
5	Pasture	41.67	41.67
	Breast-cancer	72.16	74.23
	Vehicle	72.92	73.26
	Eucalyptus	21.20	24.00
	Audit	96.32	97.21

learning was another domain of RD [12]. Finally, in social networks, [22] used RD to understand the origin of social norms and dominant behavioral and cultural trends in social networks.

Table 6 Third experiments set results

Scenario	Wins	Losses	Ties
4	6	3	6
2	4	2	9
7	4	2	9
10	4	2	9
5	4	3	8
1	4	5	6
6	3	3	9
9	3	3	9
3	3	3	9
8	2	3	10

Table 7 Third experiments set winning datasets accuracy

Scenario	Dataset	Accuracy before (%)	Accuracy after (%)
4	Squash-unstored	66.67	77.78
	Squash-stored	50.00	55.56
	Vote	96.62	97.30
	Vehicle	70.83	72.92
	Eucalyptus	20.00	34.00
	Soybean	72.41	73.28
2	Squash-unstored	61.11	77.78
	Vehicle	72.22	73.26
	Eucalyptus	24.80	28.40
	Soybean	72.41	75.00
7	Diabetes	73.18	74.33
	Vehicle	71.87	72.92
	Eucalyptus	22.80	25.20
	Audit	96.76	96.91
10	Squash-unstored	55.56	66.67
	Eucalyptus	24.40	26.80
	Audit	96.62	96.76
	Soybean	77.59	86.21
5	Squash-unstored	61.11	66.67
	White clover	61.90	66.67
	Eucalyptus	17.60	27.20
	Soybean	77.59	79.74

(continued)

Table 7 (continued)

Scenario	Dataset	Accuracy before (%)	Accuracy after (%)
1	Squash-unstored	55.56	61.11
	Squash-stored	50.00	55.56
	Eucalyptus	19.20	19.60
	Audit	96.03	97.06
6	Squash-unstored	61.11	66.67
	Eucalyptus	22.80	24.00
	Audit	96.32	96.76
9	Squash-unstored	61.11	66.67
	Eucalyptus	19.60	24.00
	Audit	96.62	96.76
3	Squash-unstored	50.00	55.56
	Eucalyptus	16.40	16.80
	Soybean	78.02	80.17
8	Eucalyptus	22.00	27.20
	Audit	96.32	96.91

Table 8 Fourth experiments set results

Scenario	Wins	Losses	Ties
5	7	3	5
10	7	5	3
9	6	4	5
1	5	6	4
2	5	6	4
3	5	7	3
4	5	7	3
6	4	3	8
7	4	5	6
8	4	5	6

7 Conclusion and Future Work

We have demonstrated in this paper how to apply *Replicator Dynamics* (RD) to boost the performance Diversified Random Forest (DRF). DRF is built using random sub-spacing with each subspace used to construct a subforest. DRF was used to impose diversity in Random Forests. Noting that subspaces may vary in their discriminative power, we applied RD to perform collective feature engineering, allowing stronger subspaces to grow and weaker ones to shrink, and potentially disappear.

As was demonstrated in Tables 2, 4, 6, and 8, favourable results were obtained demonstrating the potential of the proposed method. In our experiments, we have

Table 9 Fourth experiments set winning datasets accuracy

Scenario	Dataset	Accuracy classic RF (%)	Accuracy DRF (%)
5	Pasture	41.67	41.67
	Squash-unstored	61.11	66.67
	Vehicle	72.92	73.26
	Eucalyptus	21.20	27.20
	Credit	75.88	76.47
	Audit	96.32	96.91
	Soybean	75.86	79.74
10	Pasture	41.67	41.67
	Squash-unstored	61.11	66.67
	Breast-cancer	61.11	75.26
	Diabetes	73.56	73.95
	Eucalyptus	21.20	26.80
	Audit	96.32	96.76
	Soybean	75.86	86.21
9	Squash-unstored	61.11	66.67
	Breast-cancer	72.16	74.23
	Car	61.90	62.58
	Eucalyptus	21.20	24.00
	Audit	96.32	96.76
	Soybean	75.86	76.72
1	Breast-cancer	72.16	76.29
	Diabetes	73.56	74.71
	Car	61.90	64.28
	Vehicle	72.92	73.61
	Audit	96.32	97.06
2	Squash-unstored	61.11	77.78
	Breast-cancer	72.16	76.29
	Vehicle	72.92	73.26
	Eucalyptus	21.20	28.40
	Audit	96.32	96.91
3	Breast-cancer	72.16	73.19
	Diabetes	73.56	73.95
	Sonar	00.00	01.41
	Audit	96.32	97.06
	Soybean	75.86	80.17
4	Squash-unstored	61.11	77.78
	Eucalyptus	20.00	34.00
	Sonar	00.00	01.41
	Credit	75.88	76.47
	Audit	96.32	97.20

(continued)

Table 9 (continued)

Scenario	Dataset	Accuracy classic RF (%)	Accuracy DRF (%)
6	Squash-unstored	61.11	66.67
	Breast-cancer	72.16	73.19
	Eucalyptus	21.20	24.00
	Audit	96.32	96.76
7	Diabetes	73.56	74.33
	Eucalyptus	21.20	25.20
	Credit	75.88	76.76
	Audit	96.32	96.91
8	Squash-unstored	61.11	66.67
	Eucalyptus	21.20	27.20
	Audit	96.32	96.91
	Soybean	75.86	76.72

used a subspace factor of 2 and 4 %, a size of 500 trees for the DRF to be created, and 70 % of the features in each subspace.

In the future, we will attempt different values for these parameters. We envisage to apply the proposed method on higher dimensional datasets. RD has also the potential to address the concept drift problem in data stream ensembles, as growing and shrinking can take place according to the drifted concept.

References

1. Adeva, J.J.G., Beresi, U., Calvo, R.: Accuracy and diversity in ensembles of text categorisers. *CLEI Electron. J.* **9**(1) (2005)
2. Bache, K., Lichman, M.: Uci machine learning repository (2013)
3. Bomze, I.M.: Lotka-volterra equation and replicator dynamics: new issues in classification. *Biol. Cybern.* **72**(5), 447–453 (1995)
4. Breiman, L.: Bagging predictors. *Mach. Learn.* **24**(2), 123–140 (1996)
5. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
6. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: a survey and categorisation. *Inf. Fusion* **6**(1), 5–20 (2005)
7. Cuzzocrea, A., Francis, S.L., Gaber, M.M.: An information-theoretic approach for setting the optimal number of decision trees in random forests. In: *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pp. 1013–1019. IEEE (2013)
8. Fawagreh, K., Gaber, M.M., Elyan, E.: Diversified random forests using random subspaces. In: *Intelligent Data Engineering and Automated Learning–IDEAL 2014*, pp. 85–92. Springer (2014)
9. Fawagreh, K., Gaber, M.M., Elyan, E.: Random forests: from early developments to recent advancements. *Syst. Sci. Control Eng: Open Access J. Ibf* **2**(1), 602–609 (2014)
10. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **15**, 3133–3181 (2014)

11. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**(1), 119–139 (1997)
12. Galstyan, A.: Continuous strategy replicator dynamics for multi-agent Q-learning. *Auton. Agents Multi-agent Syst.* **26**(1), 37–53 (2013)
13. Hauert, C.: Replicator dynamics of reward & reputation in public goods games. *J. Theor. Biol.* **267**(1), 22–28 (2010)
14. Hauert, C., De Monte, S., Hofbauer, J., Sigmund, K.: Replicator dynamics for optional public good games. *J. Theor. Biol.* **218**(2), 187–194 (2002)
15. Hilbe, C.: Local replicator dynamics: a simple link between deterministic and stochastic models of evolutionary game theory. *Bull. Math. Biol.* **73**(9), 2068–2087 (2011)
16. Hofbauer, J., Sigmund, K.: Evolutionary game dynamics. *Bull. Am. Math. Soc.* **40**(4), 479–519 (2003)
17. Hutson, V., Schmitt, K.: Permanence and the dynamics of biological systems. *Math. Biosci.* **111**(1), 1–71 (1992)
18. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.* **51**(2), 181–207 (2003)
19. Lohmann, G., Bohn, S.: Using replicator dynamics for analyzing fMRI data of the human brain. *IEEE Trans. Med. Imag.* **21**(5), 485–492 (2002)
20. Maclin, R., Opitz, D.: Popular ensemble methods: an empirical study. [arXiv:1106.0257](https://arxiv.org/abs/1106.0257) (2011) (preprint)
21. Nowak, M.A., Sigmund, K.: Evolutionary dynamics of biological games. *Science* **303**(5659), 793–799 (2004)
22. Olfati-Saber, R.: Evolutionary dynamics of behavior in social networks. In: *Decision and Control, 2007 46th IEEE Conference on*, pp. 4051–4056. IEEE (2007)
23. Polikar, R.: Ensemble based systems in decision making. *Circuits Syst. Mag. IEEE* **6**(3), 21–45 (2006)
24. Roca, C.P., Cuesta, J.A., Sánchez, A.: Evolutionary game theory: temporal and spatial effects beyond replicator dynamics. *Phys. Life Rev.* **6**(4), 208–249 (2009)
25. Rokach, L.: Ensemble-based classifiers. *Artif. Intell. Rev.* **33**(1–2), 1–39 (2010)
26. Tang, E.K., Suganthan, P.N., Yao, X.: An analysis of diversity measures. *Mach. Learn.* **65**(1), 247–271 (2006)
27. Taylor, P.D., Jonker, L.B.: Evolutionary stable strategies and game dynamics. *Math. Biosci.* **40**(1), 145–156 (1978)
28. Wolpert, D.H.: Stacked generalization. *Neural Netw.* **5**(2), 241–259 (1992)
29. Yan, W., Goebel, K.F.: Designing classifier ensembles with constrained performance requirements. In: *Defense and Security*, pp. 59–68. International Society for Optics and Photonics (2004)

Research and Development in Intelligent Systems XXXII
Incorporating Applications and Innovations in Intelligent
Systems XXIII

Bramer, M.; Petridis, M. (Eds.)

2015, XIV, 410 p. 115 illus. in color., Softcover

ISBN: 978-3-319-25030-4