Dynamic Integration with Random Forests

Alexey Tsymbal[1], Mykola Pechenizkiy[2], and Pádraig Cunningham[1]

[1] Dept of Computer Science, Trinity College Dublin, Ireland
{Alexey.Tsymbal, Padraig.Cunningham}@cs.tcd.ie
[2] Dept of Math IT, University of Jyväskylä, Finland
mpechen@it.jyu.fi

Abstract. Random Forests (RF) are a successful ensemble prediction technique that uses majority voting or averaging as a combination function. However, it is clear that each tree in a random forest may have a different contribution in processing a certain instance. In this paper, we demonstrate that the prediction performance of RF may still be improved in some domains by replacing the combination function with dynamic integration, which is based on local performance estimates. Our experiments also demonstrate that the RF Intrinsic Similarity is better than the commonly used Heterogeneous Euclidean/Overlap Metric in finding a neighbourhood for local estimates in the context of dynamic integration of classification random forests.

1 Introduction

Random Forests (RF) are a relatively young (they were introduced in 2001), but effective and popular ensemble technique [5]. RF were demonstrated to compare favourably with boosting in terms of predictive performance and to be more robust with respect to overfitting noisy instances in various classification and regression domains.

In the standard RF algorithm [5] simple majority voting or averaging are used to combine the base predictions. A natural possible extension to RF is to improve the combination of trees by taking into account their local performance. One such combination technique, which could be used here, is dynamic integration (DI) [12]. In DI, local performance is estimated for each base model based on the performance on similar instances, and then this is used to calculate a corresponding weight for combining predictions with locally weighted voting, or to simply select a model with the best local performance. RF provide us with an Intrinsic Similarity metric (RFIS), which could be used in DI. The proportion of the base trees where two instances appear together in the same leaves can be used as a measure of similarity between the instances [5]. In this paper we evaluate the two alternative combination functions. We find that DI does improve the performance of RF. We also find that RFIS is very effective; this is not surprising as it is *in tune* with the dynamics of the ensemble.

This paper is organized as follows: in Section 2 we review RF, in Section 3 we consider how they can be augmented with DI, in Section 4 we present the results of our experiments, and in Section 5 we conclude with a brief summary.

J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.): ECML 2006, LNAI 4212, pp. 801 – 808, 2006.

2 Random Forests

Breiman in his paper [5] demonstrated that optimal ensemble performance could be achieved by injecting randomness in order to minimize correlation between base models while maintaining their accuracy. In RF this is achieved by combining two sources of randomness. First, instances used to grow each tree are sampled randomly without replacement from the original training set. Second, RF randomly select features at each node to grow each tree [5]. Using the Strong Law of Large Numbers, Breiman demonstrated that RF always converge so that overfitting is not a problem, that is RF never overfit as more trees are added.

RF have a set of desirable properties [5]:

(1) their predictive performance is as good as boosting and sometimes better;
(2) they are relatively robust to outliers and noise;
(3) they are faster than many other ensembles, bagging and boosting in particular;
(4) due to the use of bootstrapping, they give useful internal (so-called *out-of-bag*) estimates of error, strength (margin), correlation and feature importance;
(5) they are simple and easily parallelized.

RF were demonstrated to produce error rates not far above the Bayes rate in different application domains [5]. However, in some domains their accuracy can still be improved. For example, Robnik-Šikonja in [7] considered two ways of improving RF: (1) a combination of several feature selection criteria in order to reduce correlation in the forests, and (2) replacement of majority voting with locally weighted voting.

3 Improving Random Forests with Dynamic Integration

A number of *selection* and *combination* approaches to ensemble integration have been proposed [6, 8, 9, 12]. The most popular *combination* technique, also used in RF, is simple majority voting [1]. Weighted Voting (WV), where each vote has a weight proportional to the estimated generalization performance of the corresponding classifier, usually has better predictive performance [1].

A number of *selection* techniques have also been proposed to address the task of integration. One of the most popular and simplest selection techniques is Cross-Validation Majority (CVM), where the classifier with the highest cross-validation accuracy is selected [9].

The approaches described above are *static*. They select one model or combine the models uniformly. In *dynamic* integration information about each new instance is taken into account [7, 10, 12]. Three DI techniques based on the same local performance estimates; Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS), were considered in [12]. First, the errors of each base classifier on each instance of the training set are estimated using cross validation. This demands $O(Mn)$ additional space for saving information about the errors of M base classifiers on n training instances. The application phase begins with determining k-nearest neighbours for a new instance. After that, weighted nearest neighbour learning is used to predict the local performance of each base classifier.

Then, DS simply selects a classifier with the least local error. In DV, each base classifier receives a weight that is proportional to its estimated local performance. In DVS, the base classifiers with the errors that fall into the upper half of the error interval are discarded and DV is applied to the remaining set of classifiers.

DI was successfully applied in a number of contexts, outperforming other integration methods. In [12] DS, DV and DVS were used with bagging and boosting. In [10] DI was applied to ensembles of base classifiers generated on different feature subsets. In [8] an adaptation of the DI techniques to regression was considered.

RF have the very appealing property that each tree is built on a bootstrap replicate. The remaining (out-of-bag) instances are useful for the evaluation of the base trees' accuracy, margin, correlation, and even feature importance [1]. This property can be used in DI as well. With out-of-bag instances there is no need for cross validation or for a separate validation set.

Different distance functions can be used in DI. The simplest and most common way is to use the Euclidean distance with numeric features, and the overlap distance with categorical features, as in the heterogeneous Euclidean/overlap metric (HEOM) [13]. HEOM was demonstrated to be robust and difficult to compete with in many domains [13]. However, RF provide us with RFIS, which could be used in DI as well. The proportion of the trees where two instances appear together in the same leaves can be used as a measure of similarity between them [5]. It is important to note that two instances that are close together in the HEOM space might have relatively small RFIS if they are near the classification boundary. RFIS demands $O(nK)$ additional space for saving information about n training instances in the leaves of the K trees.

In order to calculate the weight in model i in DI for a new instance **x**, we use:

$$w_i(\mathbf{x}) = \sum_{j=1}^{k}\left(I_{OOB_i}(\mathbf{x}_j)\cdot\sigma(\mathbf{x},\mathbf{x}_j)\cdot mr_i(\mathbf{x}_j)\right) \Big/ \sum_{j=1}^{k}\left(I_{OOB_i}(\mathbf{x}_j)\cdot\sigma(\mathbf{x},\mathbf{x}_j)\right) \tag{1}$$

where k is the size of the neighbourhood, OOB_i is the set of out-of-bag instances for model i and $I()$ is an indicator function, $\sigma(\mathbf{x},\mathbf{x}_j)$ is a distance-based relevance coefficient, and $mr_i(\mathbf{x}_j)$ is the margin of model i on jth nearest neighbour of **x**. Margin can be defined as follows for a classifier with crisp outputs:

$$mr_i(\mathbf{x}) = \begin{cases} 1, & h_i(\mathbf{x}) = y(\mathbf{x}) \\ -1, & h_i(\mathbf{x}) \neq y(\mathbf{x}) \end{cases} \tag{2}$$

In fact, weight (1) represents the expected margin of model i on instance **x**. We normalize weights (1) to be non-negative and to sum to one in order to apply them in DI. In our experiments with the two distance metrics we use the inverse HEOM distance and the cube of RFIS as the corresponding distance-based weight coefficients:

$$\sigma_{HEOM}(\mathbf{x},\mathbf{x}_j) = 1/HEOM(\mathbf{x},\mathbf{x}_j) \tag{3}$$

$$\sigma_{RFIS}(\mathbf{x},\mathbf{x}_j) = RFIS^3(\mathbf{x},\mathbf{x}_j) \tag{4}$$

In our experiments we also consider a non-weighted variant of (1), demonstrating that the use of weights is usually superior for both of the distance metrics.

4 Experimental Studies

In our experiments we use an implementation based on the machine learning library WEKA 3.4.2 [14]. Information Gain is used as the splitting criterion, and the number of randomly selected features in each node is $\lfloor \log_2 M + 1 \rfloor$, where M is the number of features in the dataset.

In our experiments we use 27 benchmark datasets. 24 of these datasets are from the UCI ML repository [3]. The Parity2 and Parity3 datasets were considered in [7]. They have 2 and 3 binary parity features respectively and 10 random binary features. The Images dataset consists of 1000 image windows drawn from 2 monochrome images of natural scenes. These images were previously considered in [2]. We estimate accuracy and margin after 30 runs of hold-out cross validation with 70/30% train/test split of each dataset.

As was mentioned before, RF often give an error rate comparable to the Bayes rate. This is especially so for the relatively simple UCI datasets. Thus, it is no surprise that their accuracy is difficult to beat for any technique, including DI. In our experiments, on 12 of the 27 datasets there was a statistically significant accuracy improvement due to the use of DI. With the remaining datasets the difference in accuracy was insignificant. We continue the analysis of experimental results focusing on these 12 datasets.

The first surprising tendency we could observe was that the accuracy of DS was very poor. On most datasets DS significantly decreased the accuracy of RF with any local learning scheme. Only with 2 datasets was its accuracy better; MONK-2 and Parity2. These datasets represent artificial concepts "well suitable" for dynamic selection. Such a poor behaviour of DS is surprising, because much research in the area of ensembles is concentrated on (dynamic) classifier selection, and this is justified by its good performance in many application domains. However, in the context of RF, the base models are usually weak and diverse, which makes the task of classifier selection difficult. We continue the analysis of experimental results focusing on DV and DVS. They give close results, with DVS being a little better on average.

Naturally, accuracy with DI usually decreases with the increase in the size of neighbourhood, becoming closer to simple static majority voting. DI is not very sensitive to the size of neighbourhood. 15 and 31 instances give close results, both for the weighted and non-weighted cases, with 15 being a slightly better neighbourhood on average. We continue our analysis focusing on the size of neighbourhood equal to 15.

Now let us consider different local learning schemes for DI. In Fig. 1 the accuracy of plain RF with static voting (*SV*) is compared with RF with DVS for the 4 different local learning schemes; HEOM, equally-weighted (DVS_{HEOM}) and locally weighted (DVS_{HEOMW}), and RFIS, equally-weighted (DVS_{RF}) and locally weighted (DVS_{RFW}) for 4 different ensemble sizes (10, 25, 50 and 100), averaged over the 12 datasets.

This figure reveals a few interesting tendencies. First, it shows the average improvement due to DI, which is more than 1.5% with any local learning scheme for the ensembles with 100 trees. Second, all the schemes give close results. However, it can be seen that the locally weighed schemes out-perform their equally weighted counterparts, and RFIS results usually out-perform the corresponding HEOM results. An interesting result is that the locally weighted RFIS scheme clearly stands out on the figure. As we shall later see, this superiority will also be supported by tests for statistical significance and the analysis of classification margin for each dataset.

In Table 1 accuracy results are given for the ensembles of 100 trees for plain RF with static voting (*SV*) and for the four local learning schemes with DVS (*DVS_{HEOM}*, *DVS_{HEOMW}*, *DVS_{RF}* and *DVS_{RFW}*) for the 12 datasets. The table includes the dataset name, the minimum, average and maximum accuracy of ensemble members, and accuracies for the five integration strategies. Numbers given in bold represent the significant wins of corresponding DVS strategies over SV (according to the paired *t*-test with 0.95 level of significance).

This table demonstrates the fact that RF contain weak and highly diverse base classifiers. In many domains RF out-perform the best component decision tree (except the Glass, Zoo and Parity problems). Of the four local learning strategies, locally weighted RFIS (DVS$_{\text{RFW}}$) demonstrates the most robust behaviour with the best average accuracy and 9 wins (with 8 wins for DVS$_{\text{HEOM}}$ and 7 wins for DVS$_{\text{RF}}$ and

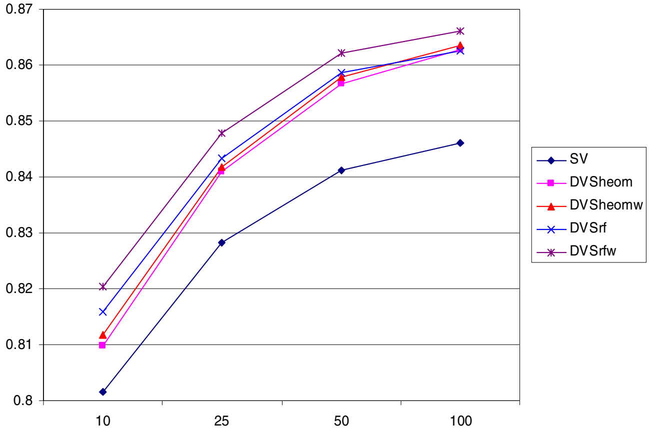

Fig. 1. Accuracy of plain and DVS RF for different local learning schemes and ensemble sizes

Table 1. Accuracy of plain RF and DVS RF for the four local learning schemes

Dataset	Min	Aver	Max	SV	DVS$_{\text{HEOM}}$	DVS$_{\text{HEOMW}}$	DVS$_{\text{RF}}$	DVS$_{\text{RFW}}$
Audiology	0.316	0.507	0.707	0.727	**0.741**	**0.740**	**0.739**	**0.739**
Car	0.755	0.830	0.888	0.935	**0.938**	**0.937**	0.936	0.937
DNAp	0.385	0.636	0.872	0.908	0.914	0.911	0.908	0.913
Glass	0.495	0.637	0.770	0.762	0.765	0.764	0.770	**0.772**
Images	0.566	0.639	0.708	0.85	**0.859**	**0.86**	**0.857**	**0.859**
MONK-1	0.624	0.824	0.989	0.997	**0.999**	**1.000**	**1.000**	**1.000**
Parity2	0.397	0.658	0.999	0.925	**0.973**	**0.974**	**0.978**	**0.977**
Parity3	0.350	0.543	0.860	0.639	**0.716**	**0.724**	**0.713**	**0.724**
Sonar	0.524	0.689	0.835	0.830	**0.841**	0.840	**0.840**	**0.844**
Tic-tac-toe	0.673	0.765	0.845	0.936	**0.960**	**0.961**	**0.961**	**0.966**
Vehicle	0.605	0.675	0.738	0.746	0.748	0.748	0.749	0.749
Zoo	0.709	0.844	0.959	0.898	0.898	0.904	0.899	**0.912**
Average	*0.533*	*0.687*	*0.848*	*0.846*	*0.863*	*0.864*	*0.863*	*0.866*

$\text{DVS}_{\text{HEOMW}}$). DI (DVS strategy) always gives similar or better accuracy than SV in these domains. The same situation holds true with DV and with the ensembles of other sizes (10, 25 and 50).

Besides the accuracy for the different integration techniques considered we also measured *classification margin* for SV, DV and DVS. The margin of a classifier *h* on instance **x** can be measured as the extent to which the average vote for the right class *y(***x***)* exceeds the maximal average vote for any other class [5]. Average margin over the test instances represents an estimate of expected margin for the classification problem considered and is an important characteristic for any learning algorithm [5,7].

In Table 2 margin is given for plain RF and for the 4 local learning schemes with DVS. From this table one can see that DI always increases the margin of plain RF on these 12 datasets. Interestingly, this increase is always significant. Besides, DI often increased the margin even when the accuracy of DI remained the same with SV (on the rest of 27 datasets). This behaviour is not so surprising, as the notion of a diverse ensemble is somewhat at odds with the concept of a high margin, i.e. diversity can be achieved by *squeezing* the margin.

Interestingly, the margins of weighted schemes are always greater than the corresponding non-weighted margins, and the margins using RFIS are always greater than the corresponding HEOM margins. Another interesting and somewhat surprising tendency when one considers separate local learning schemes is that while all the other three schemes usually give pretty close results, the margin with locally weighted RFIS, DVS_{RFW}, usually clearly stands out and is always statistically significantly higher than all the other corresponding margins, supporting its relative superiority in accuracy shown in Table 1. The results in Table 2 clearly show the superiority of RFIS over HEOM in finding a neighbourhood for DI. The fact that the locally weighted RFIS always produces a statistically significantly greater margin, even though the corresponding accuracy may not always be significantly different in comparison with the other local learning schemes, demonstrates its greater strength in this context.

Table 2. Classification margin for plain RF and for the four local learning schemes with DVS

Dataset	SV	DVS_{HEOM}	$\text{DVS}_{\text{HEOMW}}$	DVS_{RF}	DVS_{RFW}
Audiology	0.255	0.291	0.302	0.296	0.314
Car	0.701	0.729	0.733	0.735	0.754
DNAp	0.267	0.298	0.302	0.310	0.322
Glass	0.370	0.386	0.394	0.392	0.411
Images	0.276	0.287	0.289	0.289	0.295
MONK-1	0.614	0.689	0.700	0.716	0.754
Parity2	0.315	0.434	0.454	0.449	0.484
Parity3	0.092	0.160	0.172	0.165	0.187
Sonar	0.377	0.397	0.406	0.406	0.420
Tic-tac-toe	0.513	0.571	0.575	0.582	0.608
Vehicle	0.418	0.426	0.427	0.429	0.434
Zoo	0.752	0.761	0.775	0.763	0.782
Average	*0.413*	*0.452*	*0.461*	*0.461*	*0.480*

In our experiments, we considered *two bias/variance decompositions*; those of Kohavi and Wolpert [6] and Breiman [4]. They closely capture the original squared loss definitions and have a behaviour that corresponds with intuition. Analysing the behaviour of DS, we could see that DS tries to reduce bias at the expense of the considerable increase in variance. The increase in variance was huge, and on some datasets bias was increased too. DV and DVS reduce error by reducing bias while trying to keep variance the same. DV and DVS, on these datasets, always decrease bias and this decrease is always significant. Sometimes this is accompanied by an insignificant increase in variance. Comparing DV and DVS, we could see that DVS, as a technique involving classifier selection, tries to further decrease bias. Interestingly, this is not always accompanied by an increase in variance, and on average the variance terms of DV and DVS are the same. More detailed experimental results for the present study including numbers for the bias/variance decomposition (which are not included here due to space limitations) are made available online as a technical report [11].

5 Conclusions

One way for improving RF is to replace majority voting with a more sophisticated combination function such as DI. Our experiments demonstrated that DI was able to improve the accuracy of RF on 12 out of 27 datasets.

More detailed experimental analysis revealed a few interesting tendencies. DV and DVS were demonstrated to always increase margin in comparison with the usual RF – a characteristic that is similar to that of boosting. Bias/variance analysis demonstrated that DV and DVS tended to decrease bias while keeping variance the same. DS was proven to be inappropriate in this context, always significantly increasing variance.

Among the distance functions and local learning schemes considered in DI, the best combination was RFIS with locally weighted learning. Interestingly, this combination usually resulted in a significantly greater margin than all the other techniques, even when accuracy remained the same. In general, the RFIS metric demonstrated very promising behaviour, and it is an interesting question for further research whether this superiority will hold true in other data mining tasks.

Acknowledgments. This material is based upon work supported by the Science Foundation Ireland under Grant S.F.I.-02/N.1/111 and by the Academy of Finland.

References

1. Bauer E., Kohavi R.: An empirical comparison of voting classification algorithms: bagging, boosting, and variants, *Machine Learning*, 36 (1,2) (1999) 105-139
2. Bingham E., Mannila H.: Random projection in dimensionality reduction: applications to image and text data. In: *Proc. 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining KDD 01*, ACM Press (2001) 245-250
3. Blake C.L., E. Keogh, C.J. Merz: UCI repository of machine learning databases [http:// www.ics.uci.edu/ ~mlearn/ MLRepository.html], Dept. of Information and Computer Science, University of California, Irvine, CA (1999)

4. Breiman L.: Bias, Variance, and Arcing Classifiers, Tech. Report 486, Statistics Dept., University of California, Berkeley, USA (1996)
5. Breiman L.: Random Forests, *Machine Learning*, 45(1) (2001) 5-32
6. Kohavi R., Wolpert D.: Bias plus variance decomposition for zero-one loss functions. In: *Proc. 13th Int. Conf. on Machine Learning*, Morgan Kaufmann (1996) 275-283
7. Robnik-Šikonja M.: Improving random forests. In: J.F. Boulicaut et al. (eds.), *Proc. 15th European Conf. on Machine Learning ECML 04*, Springer, LNCS 3201 (2004) 359-370
8. Rooney N., Patterson D., Anand S., Tsymbal A.: Dynamic integration of regression models. In: *Proc. 5th Int. Workshop on Multiple Classifier Systems MCS 04,* LNCS 3181, Springer (2004) 164-173
9. Schaffer C.: Selecting a classification method by cross-validation, *Machine Learning*, 13 (1993) 135-143
10. Tsymbal A., Pechenizkiy M., Cunningham P.: Sequential genetic search for ensemble feature selection. In: *Proc. 19th Int. Joint Conf. on Artificial Intelligence IJCAI 05*, Morgan Kaufmann (2005) 877-882
11. Tsymbal A., Pechenizkiy M., Cunningham P.: Dynamic integration with random forests. Tech Report TCD-CS-2006-23, Dept of Computer Science, Trinity College Dublin, Ireland (2006) (available online at http://www.cs.tcd.ie/publications/tech-reports/reports.06/)
12. Tsymbal A., Puuronen S.: Bagging and boosting with dynamic integration of classifiers. In: D.A. Zighed, J. Komorowski, J. Zytkow (eds.), *Principles of Data Mining and Knowledge Discovery, Proceedings of PKDD 00*, Springer, LNAI 1910 (2000) 116-125
13. Wilson D.R., Martinez T.R.: Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research,* 6(1) (1997) 1-34
14. Witten I., Frank E.: Data Mining: Practical Machine Learning Tools With Java Implementations, San Francisco: Morgan Kaufmann (2000)