Feature Selection Mechanisms for Ensemble Creation : A Genetic Search Perspective

César Guerra-Salcedo
Department of Computer Science
Colorado State University
Fort Collins, Colorado 80523 USA
(970) 491-1943
guerra@cs.colostate.edu

Darrell Whitley
Department of Computer Science
Colorado State University
Fort Collins, Colorado 80523 USA
(970) 491-5373
whitley@cs.colostate.edu

Abstract

The majority of ensemble creation algorithms use the full set of available features for its task. Feature selection for ensemble creation has not been carried out except for some work on random feature selection. In this paper we focus our attention on genetic based feature selection for ensemble creation. Our approach uses a genetic algorithm to search over the entire feature space. Subsets of features are used as input for ensemble creation algorithms. In this paper we compare boosting and bagging techniques for ensemble construction together with feature selection approaches. Also we compared the memory employed for the ensembles using the well-known C4.5 induction algorithm for ensemble construction. Our approach show more reliable ensembles with less than 50% of the total number of features employed.

Introduction

Ensembles of classifiers has shown to be very effective for case-based classification problems. Several methods for ensembles has been proposed (Dietterich 1998) with significant improvings over single-classifier techniques. However, the ensemble creation algorithms have been used with the entire set of features available.

Tin Kam Ho (Ho 1998b), (Ho 1998a) published a method for construction of classifiers ensembles based on random feature selection. Her method relies on the fact that combining multiple classifiers constructed using randomly selected features can achieve better performance in classification than using the complete set of features for the ensemble creation. Ho's method is tested using C4.5 and compared against bagged and and boosted systems which employ the whole set of features for the ensemble construction. Ho's results are impressive in accuracy gain when compared with traditional ensemble creation methods. However, Ho's research only presents a traditional majority-vote scheme for her comparisons. Our principal motivation for this research is to find accurate subsets of features (using a wrapper approach for this purpose) that could be suitable for ensemble creation. Also, we want to test Ho's technique for bagged and boosted ensembles.

In this paper we compare Ho's method of constructing decision forests with forests created using features previously selected by a genetic search engine. Also, we use the same ideas to create ensembles of table-based classifiers. We compared traditional boosting and bagging methods (using the complete set of features) with Ho's method and our method. Over the set of experiments our method show better performance and much better use of the storage space (for table-based ensembles).

The paper is organized as follows. First a background review of the material involved in the research is presented. The experimental set up and results are detailed in section 3. In section 4 a brief discussion is presented.

Background Review

Most of the time an ensemble of classifiers is more accurate than a single classifier. Two methods for ensemble construction have been widely used, Boosting (Dietterich 1998),(Quinlan 1996) (particularly AdaBoost.M1 (Freund & Schapire 1996)) and Bagging (Breiman 1994).

Random Subspace Method : RSM

The random subspace method (Ho 1998b), (Ho 1998a) for ensemble construction relies on a pseudorandom procedure that selects a subset of features (a random subspace of features) from the feature space. The instances in the dataset are projected to this subspace and a decision tree is constructed using the projected examples. There are 2^n possible feature selections that can be made. With each selection a decision tree can be constructed. Ho suggested to construct the trees forming an ensemble using a random selection of $n/2$ features from the complete set of features. Ho was able to find more accurate tree-based ensembles using RSM than those constructed using the complete set of features. However, in Ho's research neither bagging nor boosting were employed for ensembles constructed with the RSM method. To classify an unseen case x, each classifier in the ensemble votes on the class for x. The class with the most votes is the class predicted by the ensemble (majority-vote scheme).

Dataset	Features	Classes	Train Size	Test Size
LandSat	36	6	4435	2000
DNA	180	39	2000	1186
Segment	19	7	210	2100
Cloud	204	10	1000	633

Table 1: Dataset employed for the experiments. In the DNA dataset the attributes values are 0 or 1. In the Segment and the Cloud dataset the attributes values are floats. In the LandSat dataset the attribute values are integers.

Feature Subset Selection Problem

Searching for an accurate subset of features is a difficult search problem. Search spaces to be explored could be very large. In a cloud classification problem in which each cloud is defined by 204 features there are 2^{204} possible features combinations.

The use of genetic algorithms as search techniques for feature selection is not new (Bala *et al.* 1995) (Vafaie & Jong 1994) (Guerra-Salcedo & Whitley 1998) (Turney 1997). Traditionally each chromosome in the population represents a possible subset of features that is presented to the inducer. The fitness of the chromosome is based on the accuracy of the evolved subset of features to predict class values for unseen cases. However in all the references cited above the final product is a single classifier. We do not know of any application involving GA's and ensemble creation.

For the experiments reported here we combine the outputs of several runs of a GA-inducer system in one ensemble of classifiers. The GA used for our experiments is an implementation of Eshelman's CHC (Eshelman 1991).

As part of this research we employed C4.5 (Quinlan 1993) and a table-based classifier called Euclidean Decision Tables (EDT) (Guerra-Salcedo & Whitley 1998).

Setups and Results

A series of experiments were carried out using publicly available datasets provided by the Project Statlog [1] the UCI machine learning repository (C. Blake & Merz 1998) and by Richard Bankert of the Naval Research Laboratory. Table 1 shows the datasets employed for this research.

Ensemble Related Setups

Our main objective is to compare the accuracy of ensembles constructed using three different methods for feature selection: First, features selected using a genetic algorithm. Second, features selected using RSM. Third, ensembles constructed using the complete set of features available. For each method four ensemble creation schemes were used.

- Simple ensemble creation in which an ensemble is formed by classifiers and trained with the complete set of training elements. For this approach a majority-class voting scheme is used for class prediction.
- Bagged ensemble creation using the Bagging algorithm described in previous sections.
- Two versions of AdaBoost.M1 called AdaBoost.M1.1 and AdaBoost.M1.2 (as described in (Guerra-Salcedo & Whitley 1999)).

GA-Classifier Setup

Using each dataset original training file, 50 independent train and test files were randomly generated $((Tr_1, Ts_1), (Tr_2, Ts_2), .., (Tr_{50}, Ts_{50}))$. 50 different experiments using CHC as search engine were run using these files; experiment i used files (Tr_i, Ts_i). For a particular experiment i a chromosome represents a plausible feature selection vector. In order to evaluate a chromosome and obtain its fitness, a classifier C_i was constructed using Tr_i and tested using Ts_i. After 10000 trials the best individual was saved. At the end of each set of experiments (one for each dataset), 50 individuals were saved. Each one of those individuals was meant to be used as a feature template for a classifier in a particular ensemble. For an ensemble E_k (k representing the dataset employed for that ensemble) the classifier $C^k{}_j$ uses the individual j in the set of individuals saved for the experiment corresponding to k.

Results

Two different sets of results were obtained from our experiments. First, a comparison between different approaches for constructing ensembles based on feature selection methods was carried out. Ensembles were constructed using the complete set of features, a randomly generated subset of features (RSM) and a subset of features obtained by genetic search. For each feature-selection method four different methods of ensemble creation were employed; construction based on the complete set of instances, bagging, AdaBoost.M1.1 and AdaBoost.M1.2.

For RSM, 50 feature subsets were generated once for each dataset. The subsets were then used as input for the ensemble-constructor algorithm.

Effectiveness of the Methods The results presented in Table 2 and Table 3 are the average of 10 independent runs. Table 2 presents the results of ensemble creation algorithms using EDT as a constituting classifier for the ensembles. For Table 3 the base classifier for the ensembles was C4.5. In both tables, the rows labeled CHC-EDT and CHC-C45 represents ensembles constructed using the features obtained by CHC using EDT and C4.5 as evaluation functions respectively.

In Table 2 the EDT-based ensembles created using the features obtained by the genetic search approach, either CHC-EDT or CHC-C4.5, was best in 12 out of 16 experiments. CHC-EDT was better in two experiments and CHC-C4.5 was better in 11 experiments (one tie with CHC-EDT for the Satellite dataset using AdaBoost.M1.1). Also, the features obtained by

[1] ftp.ncc.up.pt: pub/statlog/datasets

Fea.Sel	Nor				Bagg				Boo.1				Boo.2			
Meth	DNA	SAT	SEG	CLD	DNA	SAT	SEG	CLD	DNA	SAT	SEG	CLD	DNA	SAT	SEG	CLD
CHC-C45	**92.7**	90.9	91.0	**79.4**	**92.7**	**91.4**	91.5	**79.8**	**94.4**	**91.2**	88.7	**79.8**	**93.8**	**91.4**	90.4	**79.5**
CHC-EDT	92.4	**91.2**	92.3	78.6	92.3	91.1	92.7	78.5	93.3	**91.2**	92.0	78.6	93.0	91.2	92.4	77.9
RSM	87.6	91.0	**94.1**	37.6	88.0	91.0	**93.8**	38.2	87.5	91.1	**93.1**	37.7	87.9	91.0	**94.1**	40.3
All Feat.	75.9	89.5	87.4	34.0	76.6	89.5	87.3	33.9	75.6	89.5	87.4	34.0	74.0	88.0	85.4	34.3

Table 2: Accuracy (% of correctly classified instances) using EDT as base classifier for the ensemble.

Fea. Sel	Nor				Bagg				Boo.1				Boo.2			
Meth	DNA	SAT	SEG	CLD	DNA	SAT	SEG	CLD	DNA	SAT	SEG	CLD	DNA	SAT	SEG	CLD
CHC-C45	94.2	90.3	91.4	**80.0**	94.2	**90.6**	93.5	**80.4**	**95.2**	91.0	95.2	81.5	95.0	91.0	**95.4**	81.5
CHC-EDT	92.6	90.1	92.1	79.9	93.0	90.5	93.3	80.2	94.3	**91.3**	93.8	**81.9**	93.9	**91.3**	93.3	**82.2**
RSM	**94.5**	**90.9**	**93.8**	77.1	93.9	90.4	**94.6**	79.6	95.1	90.8	**95.3**	80.1	**95.3**	91.2	95.2	79.4
All Feat.	92.3	85.3	91.0	69.5	**94.5**	89.3	93.7	78.6	94.8	90.9	94.4	81.3	94.8	91.0	94.2	80.7

Table 3: Accuracy (% of correctly classified instances) using C4.5 as base classifier for the ensemble.

Algorithm	DNA	SAT	SEG	CLOUD
CHC-C45	B.1	B.2/Bagg.	Bagg.	B.1/Bagg.
CHC-EDT	B.1	B.2/B.1/Nor	Bagg.	Nor/B.1
RSM	Bagg.	B.1	B.2/Nor	B.2
All Feat.	Bagg.	B.1/Nor/Bagg	Nor/B.1	B.2

Table 4: Comparison between Bagging (Bagg.), AdaBoost.M1.1(B.1), AdaBoost.M1.2(B.2) and the use of the complete set of instances (depicted as Nor) for ensemble creation using EDT as base classifier.

Algorithm	DNA	SAT	SEG	CLOUD
CHC-C45	B.1	B.2/B.1	B.2	B.1/B.2
CHC-EDT	B.1	B.2/B.1	B.1	B.2
RSM	B.2	B.2	B.1	B.1
All Feat.	B.1/B.2	B.2	B.1	B.1

Table 5: Comparison between Bagging (Bagg.), AdaBoost.M1.1 (B.1), AdaBoost.M1.2 (B.2) and the use of the complete set of instances (depicted as Nor) for ensemble creation using C45 as base classifier.

CHC-C4.5 seems to be more effective for EDT-based ensembles than the ones obtained by the CHC-EDT system itself. They were the best option in 10 out of the 12 times when the genetic-search approach was better than the other feature selection methods.

The ensembles created using the random subspace method won in four competitions. This option seems to be the best option for datasets with small number of features as in the Segmentation dataset. For EDT-based ensembles and the datasets employed in this research, the traditional use of the complete set of features for ensemble creation was least robust. This option presented the worst performance.

From the comparisons between bagging, AdaBoost.M1.1, AdaBoost.M1.2, and the use of the complete set of instances for EDT-based ensembles, bagging and AdaBoost.M1.1 were the most successful methods. The use of the complete set of instances was effective only in five experiments and AdaBoost.M1.2 in five too. These results are summarized in Table 4.

On the other hand, using C4.5 as base classifier for ensembles, the genetic search approach was better nine times (five for CHC-C4.5 and four for CHC-EDT), RSM six and using all features was best for only one experiment. These results are summarized in Table 3. Once again the use of the complete set of features was not a robust alternative.

When comparing the different methods for ensemble creation,bagging, AdaBoost.M1.1, AdaBoost.M1.2 and the use of the complete set of instances, for C4.5-based ensembles. AdaBoost.M1.1 were the most successful method with 11 experiments. AdaBoost.M1.2 were the second most successful method with 9 experiments. Bagging and the complete set of instances were the worst approaches with zero successful experiments each one. These results are summarized in Table 5.

When comparing the accuracy of EDT-based ensembles with the accuracy of C4.5-based ensembles, C4.5-based ensembles were more accurate (higher accuracy percentages). The results are summarized in Table 6. Except for the DNA dataset, the best accuracies were obtained with the features produced by either CHC-EDT or CHC-C4.5 systems. However, the total difference between the results obtained by CHC-C4.5 and RSM for the DNA dataset was less than 0.1% (95.19% for CHC-C4.5 and 95.27% for RSM). The traditional approach of using all the features available as not a robust alternative at all.

Memory Usage for the Ensembles The second set of experiments carried out a comparison in the number of features selected by each approach. In a table-based classifier each feature is represented as a column. Reducing the number of features reduces the number of columns as well; less columns employed in a classifier represents less memory used for storage.

An important advantage of the genetic search method for EDT is its ability to obtain small feature subsets. Smaller tables are easier to implement and to store. In our research we obtain smaller tables using the genetic search method. The comparison between the number of features obtained by the genetic-search method and the number of features obtained by the other methods is presented in Table 7. The percentage of savings in feature space for DNA are in the order of 88% compared with RSM and 93% compared to ensembles constructed using the whole set of features. For LandSat dataset the percentage of savings in feature space was

Algorithm	DNA	SAT	SEG	CLOUD
Normal	C4.5 (94.5% RSM)	EDT (91.2% CHC-EDT)	EDT (94.1% RSM)	C4.5 (80.0% CHC-C4.5)
Bagging	C4.5 (94.5% All)	EDT (**91.4% CHC-EDT**)	C4.5 (94.6% RSM)	C4.5 (80.4% CHC-C4.5)
Boosting.1	C4.5 (**95.2% CHC-C4.5**)	C4.5 (91.3% CHC-C4.5)	C4.5 (95.3% RSM)	C4.5 (81.9% CHC-EDT)
Boosting.2	C4.5 (**95.3% RSM**)	EDT (**91.4% CHC-EDT**)	C4.5 (**95.4% CHC-C4.5**)	C4.5 (**82.2% CHC-EDT**)

Table 6: Comparison Between EDT and C4.5 for ensemble creation using four different ensemble creation algorithms. The best accuracies as well as the system that produce them are depicted in parenthesis.

40% compared to RSM and 70% compared to the use of the complete set of features for the ensemble construction. For the Segmentation dataset the percentages of savings compared to RSM was 64% and 81% compared to ensembles created using all the available features. For the Segmentation dataset the percentages of savings compared to RSM was 71% and 85% compared to ensembles created using all the available features.

To have an idea of the consumption of memory when the feature selection methods are applied to decision forest, we present the results for the larger datasets (DNA and Cloud). The calculation is computed in terms of node usage for pruned trees. Table 8 show the average number of nodes per tree in the ensemble, the total number of nodes in the ensemble, the accuracy of the pruned forest and the accuracy of the ensemble divided by the total number of nodes. The best method (the one which produces the most compact decision forest) was the use of all available features. However, this method was not the best in performance. Comparing RSM and genetic-based feature selection method, RSM produced most compact forest for Satellite and Cloud datasets. The genetic-based feature selection method produced most compact forest for DNA and segmentation datasets.

Method	Av Fe DNA	Av Fe SAT	Av Fe SEG	Av Fe Cloud
CHC-C4.5	18.71	12.60	4.50	29.34
CHC-EDT	11.24	12.62	3.60	44.79
RSM	90.00	18.00	10.00	102
All Feat.	180.00	36.00	19.00	204

Table 7: Average number of features used for the ensemble.

Discussion

Ensemble construction is a very important method for improving classifier accuracy. We are proposing a novel method for selecting features for ensemble construction. Our method has been empirically shown to be more accurate for ensemble creation than other methods proposed elsewhere (Ho 1998b), (Ho 1998a). One of the advantages of our approach is the enormous percentage of savings in storage for table-based ensembles.

Also we have extended Ho's research in ensemble creation by applying her method with bagging and boosting approaches to decision forests creation.

F. Selec. Algrthm.	Creation Method	Avg. Prun.	Total Nodes	Accrcy. Prun.	Ratio Acc/Total
CHC-C4.5	Normal	82.08	4104.00	94.18	0.022948
	Bagging	109.32	5466.00	94.18	0.017230
	Boost. 1	200.60	10030.00	95.19	0.009491
	Boost. 2	202.32	10116.00	95.02	0.009393
CHC-EDT	Normal	61.95	3097.50	92.58	0.029889
	Bagging	75.95	3797.50	93.00	0.024490
	Boost. 1	117.27	5863.50	94.26	0.016076
	Boost. 2	115.55	5777.50	93.92	0.016256
RSM	Normal	268.60	13430.00	94.51	0.007037
	Bagging	229.60	11480.00	93.38	0.008134
	Boost. 1	237.19	11859.50	95.10	0.008019
	Boost. 2	232.39	11619.50	95.27	0.008199
All Features	Normal	139.00	6950.00	92.32	0.013283
	Bagging	124.51	6225.50	94.51	0.015181
	Boost. 1	166.27	8313.50	94.77	0.011400
	Boost. 2	168.91	8445.50	94.81	0.011226

F. Selec. Algrthm.	Creation Method	Avg. Prun.	Total Nodes	Accrcy. Prun.	Ratio Acc/Total
CHC-C4.5	Normal	136.55	6827.50	80.00	0.011717
	Bagging	112.32	5616.00	80.40	0.014316
	Boost. 1	135.44	6772.00	81.53	0.012039
	Boost. 2	132.88	6644.00	81.53	0.012271
CHC-EDT	Normal	152.03	7601.50	79.89	0.010510
	Bagging	122.27	6113.50	80.20	0.013119
	Boost. 1	140.91	7045.50	81.93	0.011629
	Boost. 2	139.63	6981.50	82.24	0.011780
RSM	Normal	135.63	6781.50	77.14	0.011375
	Bagging	107.36	5368.00	79.59	0.014827
	Boost. 1	126.55	6327.50	80.10	0.012659
	Boost. 2	123.59	6179.50	79.38	0.012846
All Features	Normal	125.00	6250.00	69.48	0.011117
	Bagging	101.44	5072.00	78.57	0.015491
	Boost. 1	119.55	5977.50	81.32	0.013604
	Boost. 2	117.59	5879.50	80.71	0.013727

Table 8: Average number of nodes used for the ensemble using pruned trees. DNA dataset (top) and Cloud dataset (bottom). The Ratio column represents the Accuracy divided by the total number of nodes. F. Selec. column represents the feature selection algorithm employed.

Acknowledgments

César Guerra-Salcedo is a visiting researcher at Colorado State University supported by CONACyT under registro No. 68813 and by ITESM.

References

Bala, J.; Jong, K. D.; Huang, J.; Vafaie, H.; and Wechsler, H. 1995. Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification. In *14th Int. Joint Conf. on Artificial Intelligence (IJCAI).*

Breiman, L. 1994. Bagging Predictors. Technical Report 421, Dept. of Statistics Technical Report 421, University of California, Berkeley, California.

C. Blake, E. K., and Merz, C. 1998. UCI repository of machine learning databases.

Dietterich, T. G. 1998. An experimental comparison of

three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning (submitted)* 3:1–22.

Eshelman, L. 1991. The CHC Adaptive Search Algorithm. How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In Rawlins, G., ed., *FOGA -1*, 265–283. Morgan Kaufmann.

Freund, Y., and Schapire, R. E. 1996. Experiments with a new boosting algorithm. In Saitta, L., ed., *Proceedings of the Thirteenth International Conference on Machine Learning*, 148–156. Morgan Kaufmann.

Guerra-Salcedo, C., and Whitley, D. 1998. Genetic Search For Feature Subset Selection: A Comparison Between CHC and GENESIS. In *Proceedings of the third annual Genetic Programming Conference.* Morgan Kaufmann.

Guerra-Salcedo, C., and Whitley, D. 1999. Genetic Approach to Feature Selection for Ensemble Creation. In Banzhaf; W., Daida, J.; Eiben, A. E.; Garzon, M. H.; Honavar, V.; Jakiela, M.; and Smith, R. E., eds., *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference, July 13-17.* Morgan Kaufmann.

Ho, T. K. 1998a. C4.5 Decision Forest. In *Proceedings of the 14th International Conference on Pattern Recognition*, 605–609.

Ho, T. K. 1998b. The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20-8:832–844.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers.

Quinlan, J. R. 1996. Bagging, boosting, and C4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 725–730. AAAI Press/MIT Press.

Turney, P. 1997. How to Shift Bias: Lessons from the Baldwin Effect. *Evolutionary Computation* 4(3):271–295.

Vafaie, H., and Jong, K. A. D. 1994. Improving a Rule Learning System Using Genetic Algorithms. In *Machine Learning: A Multistrategy Approach.* Morgan Kaufmann. 453–470.