Comparison of Single and Ensemble Classifiers in Terms of Accuracy and Execution Time

M. F. AMASYALI
Yildiz Technical Uni. Computer
Eng. Dept. 34349 Istanbul, Turkey
mfatih@ce.yildiz.edu.tr

O. K. ERSOY
Purdue University, School of Electrical
and Computer Engineering, Indiana,
47907, USA ersoy@purdue.edu

Abstract—**Classification accuracy and execution time are two important parameters in the selection of classification algorithms. In our experiments, 12 different ensemble algorithms, and 11 single classifiers are compared according to their accuracies and train/test time over 36 datasets. The results show that Rotation Forest has the highest accuracy. However, when accuracy and execution time are considered together, Random Forest and Random Committees can be the best choices.**

Keywords: committees of learners, mixture of experts, classifier ensembles, multiple classifier systems, consensus theory, base learners

I. Introduction

Classifiers are designed to learn a mapping function from the sample features to the sample labels. Combining classifiers is also a very popular research area known under different names in the literature such as committees of learners, mixture of experts, classifier ensembles, multiple classifier systems, and consensus theory [1]. The basic idea here is to use more than one classifier and combine the classification results, in the hope that the accuracy will be better. The key to the success of these algorithms is that they build a set of diverse classifiers (base learners).

In the literature, there are several studies on the comparison of classification accuracies of the ensemble algorithms [2,3]. In this work, we used a big dataset and algorithm collection. We also compared the execution times of the algorithms and investigated the similarities of the algorithms and the datasets.

In Sections 2 and 3, the algorithms and the datasets used in this study are presented. In Section 4, the classification accuracies are compared. In Section 5, the training and testing times are compared. The algorithm and dataset similarities are discussed in Section 6. Conclusions are given at the last section.

II. Algorithms

In this study, 12 single and 11 ensemble classification algorithms were used. In this section, these algorithms are presented.

A. Single Algorithms

In Table 1, the used single algorithms and their abbreviations are given.

TABLE I. Single Algorithms

Algorithm Name	Abbreviation	Reference
Zero Rule	ZR	-
Naïve Bayes	NB	[4]
Support Vector Machines	SMO	[5]
One Nearest Neighbor	KNN	[6]
C4.5 Decision Tree	J48	[7]
Functional Trees	FT	[8]
Random Tree	RT	[9]
Fast decision tree learner	REPT	[10]
Classification and Regression Trees	CART	[10]
Best First Tree	BFT	[11]
Alternating Decision Tree	LADT	[12]
Naïve Bayes Tree	NBT	[13]

B. Ensemble Algorithms

In Table 2, the used ensemble algorithms and their abbreviations are given.

TABLE II. Used Ensemble Algorithms

Algorithm Name	Abbre-viation	Base Learner	Reference
AdaBoost	ADB	Decision Stump	[14]
Bagging	BG	REPT	[15]
Random Forest	RNDF	RT	[9]
Rotatiton Forest	ROTF	J48	[16]
Dagging	DG	SMO	[17]
Decorate	DEC	J48	[18]
Ensemble of nested dichotomies	END	nested dichotomies	[19]
LogitBoostAB	LB	Decision Stump	[20]
MultiBoostAB	MB	Decision Stump	[21]
Ramdom Committee	RC	RT	[9]
Random Subspace	RS	REPT	[22]

For each ensemble algorithm, The number of the base learners to combine was 100. The base learners of the

978-1-61284-922-5/11/$26.00 ©2011 IEEE

ensembles are default values in WEKA [23] environment. All the algorithms were run with the WEKA environment.

III. Used Datasets

In this study, classification accuracies and execution times of 23 algorithms are compared over 36 datasets. The datasets appear in Table 3. All the datasets are from UCI repository [24]. Some of them (having discrete features) were modified by discrete to numeric transformation. Each categorical feature was replaced by s binary features encoded numerically as 0 and 1, where s is the number of possible categories of the feature.

TABLE III. Characteristics of the 36 datasets

Dataset name	The number of features	The number of classes	The number of Samples
abalone	11	19	4153
anneal	63	4	890
audiology	70	5	169
autos	72	5	202
balance-scale	5	3	625
breast-cancer	39	2	286
breast-w	10	2	699
col10	8	10	2019
colic	61	2	368
credit-a	43	2	690
credit-g	60	2	1000
d159	33	2	7182
diabetes	9	2	768
glass	10	5	205
heart-statlog	14	2	270
hepatitis	20	2	155
hypothyroid	32	3	3770
ionosphere	34	2	351
iris	5	3	150
kr-vs-kp	40	2	3196
labor	27	2	57
letter	17	26	20000
lymph	38	2	142
mushroom	113	2	8124
primary-tumor	24	11	302
ringnorm	21	2	7400
segment	19	7	2310
sick	32	2	3772
sonar	61	2	208
soybean	84	18	675
splice	288	3	3190
vehicle	19	4	846
vote	17	2	435
vowel	12	11	990
waveform	41	3	5000
Zoo	17	4	84

IV. Comparison of Classification Accuracies of Algorithms

Classification accuracies were compared by algorithms' averaged rank orders, ranking test, and pair-wise comparison using a "t-test".

For obtaining averaged rank orders of algorithms, 5×2 cross validations [25] were performed for each dataset and algorithm. In this methodology, the dataset is randomly divided into two halves. One half is used in training and the other in testing and vice versa. This validation is repeated 5 times. As a result of this validation, 10 estimates of testing accuracy were obtained for each algorithm and each dataset. We used the average of these accuracies as the performance of an algorithm over a dataset. Since the classification accuracies vary significantly from dataset to dataset, ranking methods provide a more fair comparison [26]. For each dataset, the performances of ensembles are ranked from 1 (the best) to 23 (the worst). Then all ranks are averaged over all datasets for each algorithm.

The ranking test ranks the algorithms according to the total significant wins and significant losses against the other algorithms. Each algorithm is compared with all the other algorithms (23-1=22 algorithms) over each datasets (36 datasets). So each algorithm has 22*36=792 comparisons. The ranking test count is the difference between the number of significant wins and the number of significant losses. The significances is determined by using a "t-test".

In Table 4, the averaged rank values and ranking test counts of ensemble algorithms are given. Smaller values show better performances for the averaged rank values. The bigger values show the better performances for the ranking test counts.

TABLE IV. Algorithms' Average Ranks and Ranking Test Counts

Algorithm Abbreviation	Average Rank	Ranking Test Count
ZR	21.9130	-719
NB	12.7391	-148
SMO	10.2174	-14
KNN	16.4348	-133
J48	14.2174	24
FT	12.4348	50
RT	18.5217	-135
REPT	15.3478	-4
CART	13.1304	40
BFT	13.6957	36
LADT	10.8696	123
NBT	14.0435	61
ADB	13.5652	-246
BG	8.5217	141
RNDF	5.3913	227
ROTF	4.1304	289
DG	12.5652	-127
DEC	7.8261	176
END	11.5652	126
LB	8.3043	179
MB	13.3913	-294
RC	6.9130	228
RS	10.2609	120

The 6 best performed algorithms according to their averaged ranks are Rotation Forest (ROTF), Random Forest (RNDF), Random Committees (RC), Decorate (DEC), Bagging (BG), Logit Boost (LB) from best to worst.

The best performed 6 algorithms according to their ranking test counts are ordered as Rotation Forest (ROTF), Random Committees (RC), Random Forest (RNDF), Logit Boost (LB), Decorate (DEC), Bagging (BG) from best to worst.

All of the 6 best performed algorithms are ensemble algorithms. The best performed single algorithm is Alternating Decision Tree (LADT) according to averaged rank and ranking test count.

The results of comparison of the best 6 algorithms with each other using t-test are shown in Table 5. The results are given in X(Y) form, which means the algorithm in the corresponding column has better results at X datasets out of 36 than the algorithm in the corresponding row. The number in brackets (Y) represents the number of significant wins for the column with regard to the row. A 0 means that the scheme in the corresponding column did not score a single (significant) win with regard to the scheme in the row. For example, ROTF algorithm has better result than BG with 32 datasets, and the differences with 8 out of 32 datasets are significant.

TABLE V. T-PAIR TEST RESULTS OF 6 BEST PERFORMING ALGORITHMS

	BG	**RNDF**	**ROTF**	**DEC**	**LB**	**RC**
BG	-	29(7)	32(8)	25(3)	20(5)	28(6)
RNDF	7(0)	-	25(6)	11(0)	8(2)	15(1)
ROTF	4(0)	10(0)	-	3(0)	9(0)	10(0)
DEC	11(1)	25(3)	32(6)	-	17(4)	25(4)
LB	15(4)	27(4)	27(5)	19(3)	-	25(4)
RC	8(0)	20(0)	26(5)	11(1)	10(2)	-

It can be easily seen that Rotation Forest has no significant loss against 5 other algorithms.

Rotation Forest is the best performing algorithm according to the results at Table 4 and 5. The idea of Rotation Forest is to achieve simultaneously individual accuracy and diversity within the ensemble [16].

V. COMPARISON OF EXECUTION TIMES OF ALGORITHMS

In classification applications, speed is another important criterion in addition to accuracy. In this section, 23 algorithms are compared in terms of training and testing times. In Table 6, training and testing times of all the algorithms over the largest 3 datasets are given.

TABLE VI. TRAINING / TESTING TIMES OF THE ALGORITHMS ON 3 DATASETS (THE VALUES ARE IN SECONDS)

Algorithm	**letter**	**mushroom**	**splice**
ZR	0.01/0.09	0.00/0.03	0.00/0.01
NB	0.09/3.46	0.20/1.02	0.16/1.25
SMO	40.52/1.17	3.64/0.06	7.20/0.06
KNN	0.00/21.85	0.00/19.41	0.00/31.67
J48	1.84/0.17	0.65/0.03	0.96/0.01
FT	682.7/228.71	14.86/0.09	35.79/6.86
RT	0.29/0.06	0.07/0.04	0.11/0.02
REPT	0.52/0.05	0.40/0.03	0.55/0.01
CART	12.80/0.04	4.56/0.01	5.76/0.01
BFT	27.63/0.05	3.45/0.01	5.23/0.01
LADT	1064.09/0.20	139.43/0.02	144.21/0.01
NBT	247.59/12.03	169.94/0.25	620.42/0.39
ADB	0.17/0.07	20.49/0.05	2.06/0.02
BG	36.98/1.13	30.86/0.06	55.49/0.04
RNDF	25.94/2.62	6.66/0.20	7.90/0.27
ROTF	289.57/69.86	250.6/348.7	324.2/442.7
DG	330.54/12.14	3.17/0.53	4.57/0.55
DEC	2160.04/3.51	200.82/0.08	587.15/0.05
END	525.17/222.5	0.79/0.66	4.01/0.64
LB	266.90/1.71	48.20/0.08	66.84/0.05
MB	0.18/0.05	20.04/0.04	2.05/0.01
RC	27.12/2.87	7.01/0.22	9.15/0.30
RS	27.74/3.25	20.76/1.24	37.60/0.89

Among the 6 best performing algorithms (DEC, ROTF, LB, BG, RC, and RNDF), DEC and ROTF are the slowest algorithms over 3 datasets in term of training time. ROTF also needs very long testing time.

As a result, RNDF and RC algorithms can be considered as the best algorithms when accuracy and execution time are considered together.

VI. SIMILARITIES OF ALGORITHMS AND DATASETS

The hierarchical clustering method was used to determine the similarities of the algorithms / datasets. In Figure 1, the similarities of the algorithms are shown. To compute algorithm similarities, each algorithm was considered as a point having 36 (the number of datasets) dimensions. Each dimension of a algorithm correspond to a performance of algorithms over a datasets. Then, the similarities of 23 points (algorithms) were calculated using Euclidian distance metric. After the similarity values were obtained, the hierarchical clustering process was applied to have similar hierarchical groups.

In Figure 2, the similarities of the datasets are shown. To compute dataset similarities, each dataset was thought as a point having 23 (the number of algorithms) dimensions. Each dimension of a dataset corresponds to the performance obtained by an algorithm with the datasets.

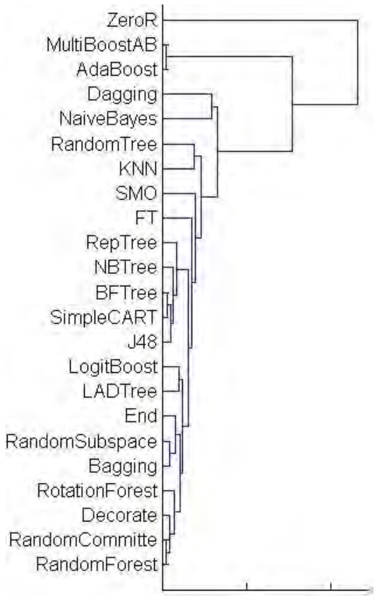

Figure 1. Similarities of Algorithms

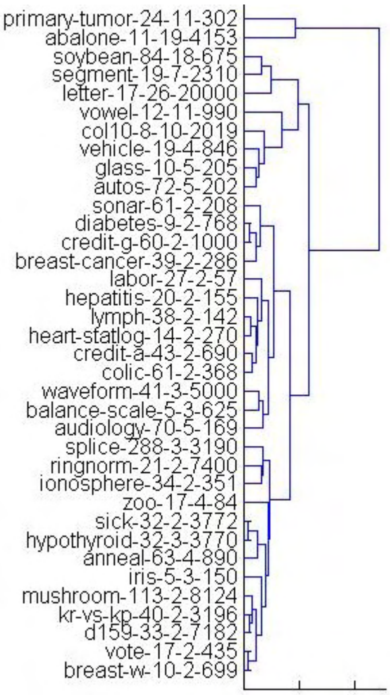

Figure 2. Similarities of Datasets

According to Figure 1, ensemble algorithms are generally grouped together.

According to Figure 2, the following conclusions are reached:

-The most similar dataset pairs have similar sample, feature and class numbers.

-Generally, datasets are grouped together according to their class numbers.

VII. Conclusions

In this study, 12 single classifiers and 11 classifier ensembles were compared over 36 datasets according to classification accuracy and execution time. The following conclusions are reached:

- The best 6 algorithms are ordered as Rotation Forest, Random Committees, Random Forest, Logit Boost, Decorate, Bagging from best to worst, according to classification accuracy.

- When accuracy and execution time are considered together, Random Forest and Random Committees are the best choices.

- When the algorithms are hierarchically grouped, the ensemble algorithms are also grouped together.

- When the dataset are hierarchically grouped, the datasets are also grouped together according to their class numbers.

As a future work, the effects of using different base learners within the ensemble algorithms on classification accuracy and execution time can be investigated.

References

[1] L.I. Kuncheva, C.J. Whitaker, “Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy”, Machine Learning, Volume 51 Issue 2, 2003.

[2] R.E. Banfield, L.O. Hall, K.W. Bowyer, D.Bhadoria, W.P. Kegelmeyer, and S.Eschrich, “A Comparison of Ensemble Creation Techniques”, MCS 2004, LNCS 3077, pp. 223–232, 2004.

[3] R.E. Banfield, L.O. Hall, K.W. Bowyer, and W.P. Kegelmeyer, “A Comparison of Decision Tree Ensemble Creation Techniques”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 1, January 2007.

[4] George H. John, Pat Langley, “Estimating Continuous Distributions in Bayesian Classifiers”, Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995.

[5] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, “Improvements to Platt's SMO Algorithm for SVM Classifier Design”, Neural Computation, 13(3) p:637-649, 2001.

[6] D. Aha, D. Kibler, “Instance-based learning algorithms”, Machine Learning, 6:37-66, 1991.

[7] R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[8] N. Landwehr, M. Hall, and E. Frank, “Logistic model trees” Machine Learning, 59(1-2):161-205, 2005.

[9] L. Breiman, “Random Forests”, Machine Learning, 45(1):5-32, 2001.

[10] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth International Group, Belmont, California, 1984.

[11] H. Shi. Best-first decision tree learning. Hamilton, NZ, 2007.

[12] G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, M. Hall, “Multiclass alternating decision trees”, ECML, 161-172, 2001.

[13] R. Kohavi, “Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid”, Second International Conference on Knowledge Discovery and Data Mining, 202-207, 1996.

[14] Y. Freund, R.E. Schapire, “Experiments with a new boosting algorithm”, Thirteenth International Conference on Machine Learning, San Francisco, 148-156, 1996.

[15] L. Breiman, “Bagging predictors”, Machine Learning, 24(2):123-140, 1996.

[16] J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, “Rotation Forest: A new classifier ensemble method”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(10):1619-1630, 2006.

[17] K.M. Ting, I.H. Witten, “Stacking Bagged and Dagged Models”, Fourteenth international Conference on Machine Learning, San Francisco, CA, 367-375, 1997.

[18] P. Melville, R.J. Mooney, “Constructing Diverse Classifier Ensembles Using Artificial Training Examples”, Eighteenth International Joint Conference on Artificial Intelligence, 505-510, 2003.

[19] L. Dong, E. Frank, S. Kramer, “Ensembles of Balanced Nested Dichotomies for Multi-class Problems”, PKDD, 84-95, 2005.

[20] J. Friedman, T. Hastie, R. Tibshirani, “Additive Logistic Regression: a Statistical View of Boosting”, Stanford University, 1998.

[21] G.I. Webb, “MultiBoosting: A Technique for Combining Boosting and Wagging”. Machine Learning. Vol.40:2, 2000.

[22] T.K. Ho, “The Random Subspace Method for Constructing Decision Forests”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8):832-844, 1998.

[23] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[24] C.L. Blake and C.J. Merz, UCI repository of machine learning databases, 1998.

[25] E. Alpaydin, “Combined 5 × 2 cv F test for comparing supervised classification learning algorithms”, Neural Computation, Volume 11 Issue 9, 1999.

[26] J. Demsar, “Statistical comparison of classifiers over multiple data sets”, Journal of Machine Learning Research, 7:1-30, 2006..