THE ASTROPHYSICAL JOURNAL, 548:219–223, 2001 February 10

ENSEMBLES OF CLASSIFIERS FOR MORPHOLOGICAL GALAXY CLASSIFICATION

D. BAZELL
Eureka Scientific, Inc., 6509 Evensong Mews, Columbia, MD 21044-6064; bazell@home.com
AND
DAVID W. AHA
Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Code 5515, Washington, DC 20375; aha@aic.nrl.navy.mil
Received 2000 June 13; accepted 2000 October 5

ABSTRACT

We compare the use of three algorithms for performing automated morphological galaxy classification using a sample of 800 galaxies. Classifiers are created using a single training set as well as bootstrap replicates of the training set, producing an ensemble of classifiers. We use a Naive Bayes classifier, a neural network trained with backpropagation, and a decision-tree induction algorithm with pruning. Previous work in the field has emphasized backpropagation networks and decision trees. The Naive Bayes classifier is easy to understand and implement and often works remarkably well on real-world data. For each of these algorithms, we examine the classification accuracy of individual classifiers using 10-fold cross validation and of ensembles of classifiers trained using 25 bootstrap data sets and tested on the same cross-validation test sets. Our results show that (1) the neural network produced the best individual classifiers (lowest classification error) for the majority of cases, (2) the ensemble approach significantly reduced the classification error for the neural network and the decision-tree classifiers but not for the Naive Bayes classifier, (3) the ensemble approach worked better for decision trees (typical error reduction of 12%–23%) than for the neural network (typical error reduction of 7%–12%), and (4) the relative improvement when using ensembles decreases as the number of output classes increases. While more extensive comparisons are needed (e.g., a variety of data and classifiers), our work is the first demonstration that the ensemble approach can significantly increase the performance of certain automated classification methods when applied to the domain of morphological galaxy classification.

Subject headings: galaxies: fundamental parameters — methods: data analysis — methods: numerical

1. INTRODUCTION

A variety of approaches have been used to perform automatic classification of galaxies based on their morphology. Neural networks and decision trees are the two most commonly used classification methods in astronomy. With both of these methods, classification is performed by presenting an algorithm with a training data set that consists of a set of objects that have been previously labeled with a class. The algorithm then tries to produce classifications of the training set objects that agree with the predefined class labels. Once the algorithm classifications and the class labels agree to a certain level of accuracy, the learning process is halted, and the internal state of the algorithm is saved. We call this a classifier. The classifier may then be applied to a set of unlabeled objects, the test set, and it will predict the class of each object.

The neural network and decision-tree approaches to morphological galaxy classification that have been used to date all rely on using a single classifier to predict the class of an unknown object. However, ensembles of classifiers can be used to combine the predictions of several individual classifiers to produce a new classifier that often has lower classification error than the individual constituents. In this paper we examine the creation of ensembles using bootstrap aggregation (Breiman 1996) of three types of classifiers: the Naive Bayes classifier, neural networks trained with backpropagation, and a decision-tree induction algorithm.

Early work on morphological classification using neural networks was done by Storrie-Lombardi et al. (1992). They used a neural network trained with backpropagation using 13 input parameters to classify galaxies into five classes: E, S0, Sa + Sb, Sc + Sd, and Irr. Their input data of 5217 galaxies was randomly split into two groups: a training set of 1700 objects and a test set of 3517 objects. They reported a 64% classification accuracy if the highest probability output was used to represent the class and a 90% accuracy if the first or second highest probability represented the output.

Naim et al. (1995) trained a neural network using backpropagation to classify 831 galaxies from the Automatic Plate Measuring Facility survey (e.g., Maddox et al. 1990). They used one output node with a range of possible values from −5 to 10. Most of their runs used 13 input features that were derived from a set of 24 features using principal component analysis. They compared the results of the neural network classifier to the results of six human experts and reported an rms error, relative to the mean of the experts, of 1.8 Revised Hubble types. This number was comparable to the dispersion among the experts.

Owens, Griffiths, & Ratnatunga (1996) used oblique decision trees for classification of the same data and features as Storrie-Lombardi et al. (1992), again splitting the data into a 1700 object training set and 3517 object test set. They did not specify if the objects in each set corresponded exactly to those of Storrie-Lombardi et al. (1992). The overall accuracy they reported was about 63% for the single training and test sets and about 64% using a fivefold cross-validation procedure.

Several methods of creating ensembles have been studied in the machine-learning literature. *Boosting* (Freund & Schapire 1996) is a family of algorithms that works iteratively by examining the examples that were incorrectly classified in the previous iterations and including multiple instances of those examples. Certain simple randomization techniques have also been shown to work well for creating

ensembles using decision trees (Dietterich 2000) and neural networks (Opitz & Maclin 1999). All of these ensemble methods have shown classification error reductions that are typically in the 20%–30% range, although some data sets have shown up to 80% reductions.

In this paper we report initial results from using ensembles of classifiers to perform morphological classification of 800 galaxies into two to six galaxy classes. A useful galaxy classifier should be able to place an example object correctly into one of several classes. We investigate the accuracy of three classification techniques: Naive Bayes, a neural network trained with backpropagation, and a decision-tree induction algorithm, when used to classify objects into multiple output classes. The neural network and decision-tree ensembles show significant improvement in accuracy over the individual classifiers, with the decision tree showing more improvement than the neural network. The Naive Bayes classifier shows no significant change. We also find that the relative improvement of ensembles over individual classifiers decreases with increasing number of output classes.

In § 2 we discuss the different classification methods we used and give a detailed description of the bagging approach to ensembles. In § 3 we describe the data we used and the features extracted from the galaxy images. Our results are presented in § 4, and a discussion of these results is given in § 5.

2. CLASSIFICATION METHODS

We compared three automatic classification methods: Naive Bayes, neural networks trained with backpropagation, and a decision-tree induction algorithm based on C4.5 (Quinlan 1993). Both backpropagation networks and various decision-tree algorithms have been used previously for astronomical classification (Storrie-Lombardi et al. 1992; Naim et al. 1995; Salzberg et al. 1995). Although to our knowledge the Naive Bayes classifier has not been previously used for galaxy classification, it is easy to understand how it performs classification, it is easy to implement, it tends to be robust to noise in the data set, and it often works quite well even when the basic assumption of conditional independence does not hold (most of the time). See, for example, Domingos & Pazzani (1997).

The Naive Bayes classifier (Ripley 1996; Mitchell 1997) uses Bayes's rule and the assumption of conditional independence of the features to calculate the probability of a class k given a feature vector (set of attributes of an object) $\boldsymbol{x} = (x_1, \ldots x_m)$. Bayes's rule states that

$$p(k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid k)p(k)}{p(\boldsymbol{x})} . \quad (1)$$

If we assume that the elements of the feature vector $\boldsymbol{x}$ are conditionally independent, then we can rewrite $p(\boldsymbol{x} \mid k) = \prod_i p(x_i \mid k)$. This gives us the final Naive Bayes's classification rule

$$p(k \mid \boldsymbol{x}) \propto p(k) \prod_{i=1}^{m} p(x_i \mid k) . \quad (2)$$

Thus, given a new feature vector, the Naive Bayes classifier can determine the probability that the feature vector belongs to a given class. The overall normalization is taken such that the probabilities sum to 1. The algorithm we used was implemented in the Waikato Environment for Knowledge Analysis (WEKA; Witten & Frank 1999). This is a freely available software package that contains a large variety of machine-learning algorithms.[1]

Neural networks trained with backpropagation (Rumelhart et al. 1986; Hertz, Krogh, & Palmer 1991) have been used previously in the astronomical community for classification of galaxies by morphology (Storrie-Lombardi et al. 1992; Naim et al. 1995) and spectra (von Hipple et al. 1994) and for star/galaxy discrimination (Odewahn et al. 1992). Broadly speaking, neural networks consist of several layers of simple nonlinear processing units. The processing units, or nodes, are connected together via weights, giving some connections more significance than others. Each node calculates its total input, y, by computing the inner product of the inputs with the weight vector, $y = \boldsymbol{x} \cdot \boldsymbol{w}$. The output of the node is then calculated by passing it through a sigmoid function $f(y) = 1/[1 + \exp(-y)]$. The magnitude of the weights is determined by minimization of an objective function that depends on how close the predicted output of the network is to the true output based on a set of labeled training data.

Our neural network consisted of an input layer containing 14 nodes (one for each feature), 10 hidden nodes, and two to six output nodes. The network was fully connected; i.e., all input nodes were connected to all hidden nodes and all hidden nodes to all output nodes. The backpropagation software we used was NEVPROP.[2] The network was run for 80 epochs (presentations of all input examples) for two, three, and four classes, 120 epochs for five classes, and 100 epochs for six classes. The number of epochs was chosen to minimize the ensemble error (see § 4) and was within the range used in previous studies (Opitz & Maclin 1999). The use of 10 hidden nodes was based on prior experience showing this to be an adequate number. The network was trained using standard gradient descent, a learning rate of 0.01, and a momentum term of 0.9.

The decision-tree algorithm we used, J48, is the WEKA implementation of the last public release (Version 8) of C4.5 (Quinlan 1996). J48 operates by recursively splitting a training set based on feature values to produce a tree such that each example can end up in only one branch. An initial feature is chosen as the root of the tree, and the examples are split among branches based on the feature value for each example. If the values are continuous, then each branch takes a certain range of values. Then a new feature is chosen, and the process is repeated for the remaining examples. When the classification of a branch is pure, i.e., it contains only examples in a certain class, then the process is stopped for that branch. The decision of which feature to use for a given split is made by calculating the information gain for that feature and choosing to split on the feature that produces the highest information gain. The information, or entropy, is calculated as $S = -\sum_i p_i \log p_i$, where p_i is the fraction of examples reaching a branch with a attribute value i. The information gain for a given split can be calculated as $G = S(\text{before split}) - S(\text{after split})$. Details of the decision-tree induction algorithm can be found in Witten & Frank (1999).

[1] The WEKA software package can be found at http://www.cs.waikato.ac.nz/ml/weka.

[2] See the NEVPROP software, Version 4.1 Web site maintained by P. H. Goodman at http://www.scs.unr.edu/nevprop.

The process just described for inducing a decision tree works well for the training data sets, but it tends to produce trees that are overfitted to the training data and do not generalize well to new examples. Thus, the final tree is usually pruned to produce a more robust classifier. For the J48 algorithm, the final tree was pruned using "subtree raising," in which classification subtrees can be raised to replace their parent subtrees. The examples along any raised subtrees must then be reclassified. This approach, as well as other pruning options, results in smaller decision trees and increases the generalization ability of the decision tree. While we have discussed one implementation of decision-tree inducers, other implementations have been used for star/galaxy discrimination (Weir et al. 1995) and morphological galaxy classification (Owens et al. 1996).

An ensemble of classifiers can be implemented in a variety of ways. One is to train several individual classifiers whose output decisions can be combined (typically by voting or averaging) to allow classification of new inputs. Ensembles have been shown to perform better than individual classifiers in a variety of domains, (e.g., Bauer & Kohavi 1999; Opitz & Maclin 1999; Dietterich 2000). In this study we used an ensemble created by bootstrap aggregation (*bagging*; Breiman 1996). Bagging is one of the easiest ensemble methods to implement and was the only one studied here. This algorithm creates the different classifiers by training them on bootstrap replicates of the original training set. Each classifier's training set is created by randomly sampling, with replacement, N examples from the original training set, where N is the number of examples in the original training set. Some examples will appear more than once in the bootstrap replicates, while others will not appear at all. When an individual classifier is trained, its overall error may be higher than for a classifier trained on the original training set. However, because the ensemble is created by voting the predictions of each classifier for each test set example, if a plurality of the classifiers make the correct predictions, the ensemble will make the correct prediction. In this manner, the voting can overcome the increased overall error on the part of individual classifiers. Other methods of creating ensembles are reviewed in Dietterich (1997).

3. DATA PREPARATION

The list of galaxies and their classifications was taken from Naim et al. (1995). Certain galaxies were eliminated based on image quality and availability, leaving 800 of the original 834. We extracted an initial set of 22 features from the images taken from the Digitized Sky Survey, as described in Bazell (2000). Table 1 gives a brief description of the features used in this study. The cross-correlation matrix of the initial 22 features showed that a number of them were significantly correlated. In particular, the total area of the galaxy and its mean brightness had a correlation coefficient of 0.95, leading us to remove the area feature. Similarly, neighboring concentration indices, e.g., C2, C3, and C4, typically had correlation coefficients above 0.75. Thus, of the initial nine concentration indices, we kept only C3 and C6, which themselves were correlated at the 0.50 level. This reduced the initial set of 22 features by eight, leaving 14 features with a maximum cross correlation of -0.74 between C3 and r_{25}/r_{75}.

The 800 galaxies were randomly divided into 10 groups of 80, and one group was set aside as a test set while the other nine groups (720 galaxies) were combined into a training set. This process was repeated, setting aside each of the 10 groups as a test set and combining the remaining groups into the training set. This procedure is called 10-fold cross validation. For each cross-validation training/testing set, the test data was never seen by the algorithm during the training process. The random division into 10 groups was repeated five times resulting in five different 10-fold cross-validation training/testing sets.

We further created 25 bootstrapped data sets from each of the 10 cross-validation training sets. The replication steps were performed independently for each cross-validation training set. The cross-validation test sets of 80 galaxies served as test sets for the 25 bootstrapped training sets. Again, the bootstraps and the test sets had no galaxies in common. Each of the algorithms studied used exactly the same training and testing data sets.

4. CLASSIFICATION RESULTS

A summary of our results for the individual classifiers and the bagged ensembles is shown in Table 2. The number of output classes is shown in column (1). Columns (2), (4), and (6) show the test set error E_S of the individual classifiers, in percent, averaged over five sets of 10 cross-validation runs. Columns (3), (5), and (7) show the test set error E_B for bagged ensembles, averaged over five sets of 10 cross-validation runs.

A number of interesting patterns can be seen from these

TABLE 1

DESCRIPTION OF FEATURES USED IN MORPHOLOGICAL CLASSIFICATION

Feature Name	Description
Peak brightness	Maximum brightness level in the image
m_{q2q3}	Ratio of fitted slope of $I(r)$ vs. r for the second and third quartiles
Ellipticity	Ratio of the semimajor to semiminor axis length
Area	Number of pixels contained in the object
Max(rI)	Maximum value of the plot of $rI(r)$ vs. r
Asym	Comparison between original galaxy and galaxy rotated 180°
r_{25}/r_{75}	Ratio of the radii at which 25% and 75% of light is enclosed in a plot of $I(r)$ vs. r
$R_{\rm Bulge}$	Radius where $I(r)$ falls to 90% of peak value
C_3, C_6	Concentration indices for the annuli 3 and 6
Isophotal displacement	Maximum displacement of the centers of five isophotal levels
Isophotal filling factor	Area of an isophotal level relative to the area of the enclosing ellipse
$P_{\rm max}$	Maximum value of the normalized co-occurrence matrix, c_{ij}
Entropy	$-\sum_{i,j} c_{ij} \log (c_{ij})$

TABLE 2

AVERAGE TEST SET CLASSIFICATION ERRORS FOR SINGLE CLASSIFIERS, E_S, AND BAGGED ENSEMBLES, E_B

	NAIVE BAYES		BACKPROPAGATION		J48	
N_{class} (1)	E_S (2)	E_B (3)	E_S (4)	E_B (5)	E_S (6)	E_B (7)
2.....	16.02	16.28	14.32	12.88	15.15	12.81
3.....	31.72	31.35	24.98	22.02	27.80	21.45
4.....	50.02	50.22	46.95	43.02	50.63	41.42
5.....	53.55	53.70	55.05	49.82	59.15	50.50
6.....	57.12	57.55	59.53	55.20	61.50	54.05

results. Considering first the single classifiers, backprop performed best (lowest test set error) for two, three, and four classes. For five and six classes, the Naive Bayes algorithm performed slightly better than backprop. The backprop algorithm showed between 7% and 12% decrease in classification error when going from individual classifiers to bagged ensembles. J48 showed between 12% and 23% decrease in classification error. This was due to a combination of effects: J48 tended to have both a larger single classifier error and a smaller ensemble error. The Naive Bayes algorithm showed no significant change in error when comparing individual classifiers to bagged ensembles. Examining the ensemble errors for backprop and J48 we see that they both result in approximately the same classification error, within one or two percentage points, even though the percentage decrease in error was significantly more for the decision trees.

For backprop and J48, it is also interesting to consider the trend in the percentage change in error when using ensembles compared to single classifiers as a function of the number of output classes. As we increase the number of output classes, both backprop and J48 show a trend toward smaller percentage change in error.

5. SUMMARY AND DISCUSSION

Our preliminary results using ensembles of classifiers clearly show the utility of this technique for decreasing classification error when performing morphological galaxy classification, at least for this data set. A variety of techniques are available for creating ensembles, but we examined only bagging, which is one of the easiest to implement. Bagging to create ensembles of classifiers has been studied by a number of researchers, e.g., Breiman (1996), Bauer & Kohavi (1999), Opitz & Maclin (1999), and Dietterich (2000). It has been shown to be effective in decreasing classification errors for learning algorithms that are unstable (Breiman 1994). Basically, a classifier is unstable if perturbing the training set, as is done with the bootstrap approach by removing some training examples and repeating others, allows a different classifier to be built. Neural networks and decision trees are unstable in this sense, whereas the Naive Bayes classifier is stable. If the different classifiers within an ensemble perform well individually and their predictions are not correlated, then the combined outputs of the ensemble often will be more accurate than any of the individual classifiers. As expected, based on the instability of the algorithms, we saw a reduction in error rate for both the backpropagation network and the decision tree when using bagged ensembles.

There is still a large parameter space to explore for the different classification algorithms. For backpropagation networks, the learning rate and the momentum can be varied as well as the number of epochs trained. Opitz & Maclin (1999) also showed that ensembles created using networks trained with different initial weight values also performed well relative to individual classifiers. For J48, there are several parameters that can be set. The confidence value determines the degree of pruning; lower values cause more drastic pruning. Trees can be created without any pruning.

The trend of decreasing percentage change in error between single classifiers and ensembles with an increasing number of output classes was apparent with both the neural network and decision-tree classifiers, even though the two methods showed significantly different single classifier errors. This trend was not evident in the work of Opitz & Maclin (1999). However, they used a variety of data sets with different training and test set sizes and numbers of output classes. Nevertheless, even for six output classes, there was still a 7% and 12% decrease in error for backprop and J48, respectively. Further investigation is needed to understand the source and significance of this trend. This has direct bearing on the limits of applicability of the bagged ensemble method.

There are other approaches to handling multiclass problems. For example, distributed output codes were studied by Sejnowski & Rosenberg (1987), and error correcting output codes have been used by Dietterich & Bakiri (1995) and Ricci & Aha (1998). Applications of these methods to astronomical data are planned.

We are in the process of making a more detailed examination of the results presented here. We would like to understand why the decision-tree approach gives a larger decrease in error going from individual classifiers to ensembles. We also plan to examine the utility of ensembles for classification using other galaxy data sets, other classifiers, and other ensemble methods.

We thank the anonymous referee for making suggestions that helped clarify some concepts in this paper and make it accessible to a broader readership. This work was performed under the support of NASA AISRP grant NAG 5-8166 and was also supported by a grant from the Office of Naval Research.

REFERENCES

Bauer, E., & Kohavi, R. 1999, Machine Learning, 36, 105
Bazell, D. 2000, MNRAS, 316, 519
Breiman, L. 1994, Tech. Rep. 416, Dept. of Stat., Univ. of California, Berkeley
———. 1996, Machine Learning, 24, 123
Dietterich, T. G. 1997, AI Mag., 18, 97
———. 2000, Machine Learning, 40, 139
Dietterich, T. G., & Bakiri, G. 1995, J. Artif. Intell. Res., 2, 263
Domingos, P., & Pazzani, M. 1997, Machine Learning, 29, 103
Freund, Y., & Schapire, R. E. 1996, in Proc. 13th Intl. Conf. on Machine Learning, ed. L. Saitta (San Francisco: Kaufmann), 148
Hertz, J., Krogh, A., & Palmer, R. G. 1991, Introduction to the Theory of Neural Computation (Redwood City: Addison-Wesley)
Maddox, S. J., Sutherland, W. J., Efstathiou, G., & Loveday, J. 1990, MNRAS, 243, 692
Mitchell, T. M. 1997, Machine Learning, (New York: McGraw-Hill)
Naim, A., Lahav, O., Sodré, L., Jr., & Storrie-Lombardi, M. C. 1995, MNRAS, 275, 567

Odewahn, S. C., Stockwell, E. B., Pennington, R. L., Humphreys, R. M., & Zumach, W. A. 1992, AJ, 103, 318
Opitz, D., & Maclin, R. 1999, J. Artif. Intell. Res., 11, 169
Owens, E. A., Griffiths, R. E., & Ratnatunga, K. U. 1996, MNRAS, 281, 153
Quinlan, J. R. 1993, C4.5: Programs for Machine Learning (San Francisco: Morgan Kaufmann)
———. 1996, J. Artif. Intell. Res., 4, 77
Ricci, F., & Aha, D. W. 1998, in Proc. 10th European Conf. on Machine Learning, ed. C. Néellec & C. Rouveirol (Chemnitz: Springer), 280
Ripley, B. D. 1996, Pattern Recognition and Neural Networks (Cambridge: Cambridge Univ. Press)
Rumelhart, D. E., & McClelland, J. L. 1986, Parallel Distributed Processing, Volume 1 (Cambridge: MIT Press)
Salzberg, S., Chandar, R., Ford, H., Murthy, S. K., & White, R. 1995, PASP, 107, 279
Sejnowski, T. J., & Rosenberg, C. R. 1987, J. Comp. Syst., 1, 145
Storrie-Lombardi, M. C., Lahav, O., Sodré, L., Jr., & Storrie-Lombardi, L. J. 1992, MNRAS, 259, 8
von Hipple, T. Storrie-Lombardi, L., Storrie-Lombardi, M. C., & Irwin, M. 1994, MNRAS, 269, 97
Weir, N., Fayyad, U. M., Djorgovski, S. G., & Roden, J. 1995, PASP, 107, 1243
Witten, I. H., & Frank, E. 1999, Data Mining (San Francisco: Kaufman)