# Classification of Bent-Double Galaxies: Experiences with Ensembles of Decision Trees

Chandrika Kamath
Erick Cantú-Paz
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
P.O. Box 808, L-561
Livermore, CA 94551
kamath2@llnl.gov, cantupaz1@llnl.gov

February 22, 2002

## Abstract

In earlier work, we have described our experiences with the use of decision tree classifiers to identify radio-emitting galaxies with a bent-double morphology in the FIRST astronomical survey. We now extend this work to include ensembles of decision tree classifiers, including two new algorithms that we have developed. These algorithms randomize the decision at each node of the tree, and because they consider fewer candidate splitting points, are faster than other methods for creating ensembles. The experiments presented in this paper with our astronomy data show that our algorithms are competitive in accuracy, but faster than other ensemble techniques such as Boosting, Bagging, and Arcx4 with different split criteria.

## 1 Introduction

Recent work in classification indicates that significant improvements in accuracy can be obtained by growing an ensemble of classifiers and having them vote for the most popular class [3, 10, 1]. In addition to the improved accuracy, ensembles also have the potential for on-line classification of large databases that do not fit into memory [4], and can easily be parallelized [11]. In this paper, we compare the performance of several algorithms for decision-tree ensembles using a classification problem in astronomy, namely, the identification of bent-double galaxies in the FIRST survey.

The outline of this paper is as follows: In Section 2, we describe the different approaches we can use to create ensembles of decision trees. Next, in Section 3, we briefly describe the astronomical dataset and our approach to classifying bent-double galaxies. In Section 4, we compare and contrast the performance of various ensemble techniques, and finally, in Section 5, we summarize our work.

## 2 Ensembles of Decision Trees

The idea behind ensembles of classifiers is simple. Instead of creating a single classifier from a training set, we create a number of classifiers and have them vote for the most popular class. There is considerable diversity in the way in which ensembles of classifiers can be created [7]. In this section, we briefly discuss some of the more popular approaches for creating ensembles.

### 2.1 Randomizing the Training Set

In this approach, each classifier in the ensemble is generated using a different sample of the training set. This can be accomplished in several ways:

- **Bagging**: In this approach, a new sample of the training set is obtained through bootstrapping with each instance weighted equally [3]. This technique works very well for unstable algorithms such as decision trees and neural networks, where the classifier is sensitive to changes in the training set and significantly different classifiers are created for different training sets. In bagging, the results of the ensemble are obtained by using a simple voting scheme. Each classifier can be generated independent of the other, and randomization is introduced through

the random sampling used to create each sample of the training set.

- **Boosting**: In this case, a new sample of the training set is obtained using a distribution based on previous results [10]. Unlike the Bagging algorithm, which uniformly weights all the instances in the training set, Boosting algorithms adjust the weights after each classifier is created to increase the weights of misclassified instances. This essentially implies that the training sets for the classifiers have to be created in sequence, instead of in parallel, as in the case of Bagging. The different weights for the ensembles can either be directly incorporated into the classifier by working with weighted instances, or be applied indirectly by selecting the instances with a probability proportional to their weights. Further, in Boosting, the results of the ensemble are obtained by weighting each classfier by the accuracy on the training set used to build it. As a result, better classifiers have a greater contribution to the end result than the poorer classifiers. There are several variants of Boosting which differ in the way the instances are weighted, the conditions under which the algorithm stops, and the way in which the results from the ensemble are combined [1, 10].

- ArcX4: This algorithm is similar to Boosting, but instead of using weighted voting, it uses unweighted voting. It was proposed by Breiman [5], to test the hypothesis that the success of Boosting lies in the adaptive resampling, where greater weight was placed on the instances more frequently mis-classified so they appeared more frequently in the training set. If $m_i$ is the number of misclassifications of the $i$-th instance, then the probability that it will be selected in the next training set is proportional to $1 + m_i^4$. Note that ArcX4 can be considered a special case of the Boosting algorithm, with unweighted voting and probabilities determined as above.

## 2.2 Randomizing the Classifier

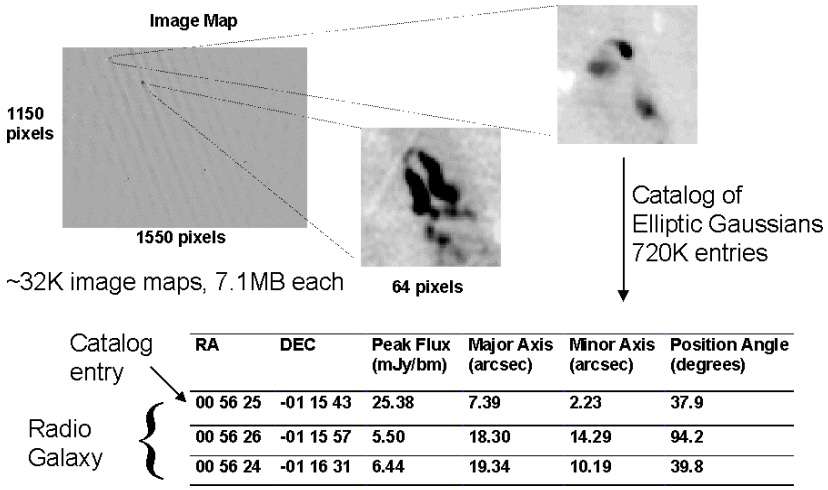
Instead of changing the input to the classifier, it is possible to create the ensemble by changing the classifier itself. For example, in neural networks, the initial weights are set randomly, thus creating a new network each time. In decision trees, instead of selecting the best split at a node, one can randomly select among the best few splits to create the ensemble [8]. Another approach would be to randomly select the features used to determine the split at each node of the tree [6].

We also consider two new approaches for introducing randomization in the classifier. These were developed as part of the Sapphire project [17]. We refer to these methods as ASPEN, for Approximate SPlitting for ENsembles (of trees):

- Sampling-based ASPEN: The first approach is based on sampling the instances at each node of the tree [12]. Instead of using all the instances at a node to make a decision, we use only a sample of the instances. As the split made at a node is likely to vary with the sample selected, this technique results in different trees which can be combined in ensembles. There are several ways in which the sampling can be implemented. For the work presented here, we use a fixed percentage of samples at each node of the tree and stop the sampling if the number of instances at a node is less than twice the number of features. We sample uniformly without replacement by throwing a biased coin for every instance to decide if the instance will be chosen or not.

- Histogram-based ASPEN: In this approach, we take a simple idea that has long been used to discretize continuous variables in decision trees, i.e. histograms, and introduce randomness in the decision [14]. Instead of sorting each of the features to determine the best split for that feature, we first create a histogram for each feature. Then, we consider the histogram bin boundaries as the potential split points. This reduces the computation involved as no sorting is needed, and instead of considering all the points midway between two consecutive sorted feature values, we consider only the bin boundaries. Next, to introduce randomness, we select an interval around the best bin boundary among all the features, and select a point randomly in this interval as the split point. In the experiments reported here, we use equal-width histograms, with the number of bins chosen to be the square-root of the number of instances at a node, and the width of the interval the same as the width of a bin.

In addition to the approaches described in the previous two sections, there are several other ways of introducing randomization in decision trees ensembles, for example by changing the features used or by randomizing the output [12]. However, in this paper, we




| RA | DEC | Peak Flux (mJy/bm) | Major Axis (arcsec) | Minor Axis (arcsec) | Position Angle (degrees) |
|---|---|---|---|---|---|
| 00 56 25 | -01 15 43 | 25.38 | 7.39 | 2.23 | 37.9 |
| 00 56 26 | -01 15 57 | 5.50 | 18.30 | 14.29 | 94.2 |
| 00 56 24 | -01 16 31 | 6.44 | 19.34 | 10.19 | 39.8 |

Figure 1: FIRST Data: Images maps and catalog entries.

will compare the two techniques proposed as the ASPEN algorithms with the following implementations of the three more popular ensemble algrithms:

- AdaBoost as proposed in AdaBoost.M1 in [10]
- Bagging as proposed in [3]
- ArcX4 as proposed in [5]

These algorithms have been implemented as part of the Sapphire software. This software also supports several different splitting criteria for making the decision at a node of the tree, such as Gini, information gain, information gain ratio, etc. [15]. In addition, we include options to create trees both with and without pruning. For the results presented here, we will consider pessimistic error pruning as described in [16].

The focus in this paper is on the comparison of several different decision tree ensemble algorithms using a specific problem in scientific data mining, namely, the classification of bent-double galaxies (see Section 3). A more extensive comparison of the two new techniques proposed in Section 2.2 is given in [12, 14]

# 3 The FIRST Dataset

We conducted our experiments with decision tree ensembles in the context of the classification of radio-emitting galaxies with a bent-double morphology in the FIRST survey. The Faint Images of the Radio Sky at Twenty-cm (FIRST) [2] is an astronomical survey using the Very Large Array at the National Radio Astronomy Observatory (http://sundog.stsci.edu). The FIRST astronomers are surveying more than 10,000 square degrees of the sky, to a flux density limit of 1.0 mJy (milli-Jansky). With the data collected through 1999, FIRST has covered about 8,000 square degrees, producing more than 32,000 two-million pixel images. At a threshold of 1 mJy, there are approximately 90 radio-emitting galaxies, or radio sources, in a typical square degree.

While radio sources exhibit a wide range of morphological types, the FIRST astronomers are particularly interested in galaxies with a bent-double morphology, as they indicate the presence of clusters of galaxies. Figure 1 has two images that were identified manually by scientists as bent-double galaxies. This visual inspection of the radio images, besides being very subjective, has also become increasingly infeasible as the survey has grown in size.

The data from the FIRST survey is available as image maps and a catalog. In Figure 1, we show an image map and the three catalog entries corresponding to one of the bent-doubles present in the image map. These large image maps are mostly "empty", that is, composed of background noise that appear as streaks in the image. The FIRST catalog [18] is obtained by processing each image map in the survey to fit two-dimensional elliptic Gaussians to each radio source. Each entry in the catalog corresponds to the information on a single Gaussian. This includes, among other things, the coordinates for the center of the Gaussian, the major and minor axes, the peak flux, and the position angle of the major axis (degrees counter-clock-wise from North).

Our approach to mining the FIRST data for bent-doubles has been described in detail elsewhere [9, 13]. Our initial focus was on the catalog data as it was

easy to work with and a good representation of all but the most complex of radio-emitting galaxies. We first grouped the catalog entries that were close to each other, and then focused on those groups that consisted of two or three catalog entries. This was based on the observation that a single entry galaxy was unlikely to be a bent-double, while four or more entries in a galaxy would make it complex enough to be of interest to astronomers. We next extracted a separate set of features for the two- and three-entry galaxies, focusing on features such as relative distances and angles between entries, that were likely to be robust and invariant to rotation, scaling, and translation. Separating the two- and three-entry galaxies enabled us to have uniform length feature vectors for each. However, it also meant that a small training set (313 examples) was split further into smaller training sets of 118 examples for two-entry and 195 examples for three-entry sources, respectively. Our focus in this work is on the three-entry sources, with the training set consisting of 167 bent-doubles and 28 non-bent-doubles. There are over 15,000 such galaxies in the data collected through 1999, making a visual inspection tedious. Note that the training set is unbalanced, with far more bent-doubles than non-bent-doubles.

Once we had extracted an intial set of features, we continued refining the features until the cross-validation error for a decision tree classifier was reduced to about 10%, a number the astronomers felt was sufficient for their use. The tree created using this set of features was then used to classify unlabeled galaxies. Several of these galaxies were shown to the astronomers for validation. Since we wanted to use this new set of galaxies to enhance our training set, we selected a higher percentage of galaxies that had been classified as non-bents. This process of validation is rather tedious and has the drawback of being not only subjective, but somewhat inconsistent as the labels assigned by an astronomer are subject to the drift common to human labelers. Therefore, we were able to validate only 290 galaxies, of which 92 were bents and 198 non-bents. We first used these newly validated galaxies as a separate testing set. Next, we combined this set with the original training set of 195 instances, and used the combined set of 485 instances to evaluate the algorithms using cross validation.

# 4 Experimental Results

We next describe the results of our experiments with various ensembles methods discussed in Section 2. There are two sets of experiments. The first set creates a single tree or an ensemble using the original 195 instances and tests it on the newly labeled 290 instances. We present the results for (1) a single tree created using exact splitting at each node, (2) a single tree created using histogram-based approximate splitting (the best bin-boundary is taken as the split point), and ensembles of 10 and 20 trees created using the (3) histogram-based ASPEN approach, (4) AdaBoost, (5) Bagging, and (6) ArcX4. Since the training set was small, we did not try the sampling-based ASPEN approach for generating the ensembles. We included the results with (2), i.e. single tree with histogram-based splitting, in order to compare it with (1), where the single tree was created with an exact split at each node using a sort. The test error rates for this set of experiments are given in Table 1. These results are obtained after 10 runs. In Bagging and histogram-based ASPEN, the randomization of the algorithm results in different test error for each run. In these cases, the standard error is included as well. All results are presented for a tree or ensemble created without pruning; since the training set was rather small and unbalanced, pruning increased the error.

The ten-fold cross validation error using the 195 instances for training a single tree was approximately 10%. The results in Table 1 indicate that if a classifier is created using these 195 instances and tested on a different set of 290 instances, the error increases almost by a factor of 3. These results must however be viewed with caution. First, the training set has far more bent-doubles (167) than non-bent-doubles (28). In contrast, the test set has more non-bents (198) than bents (92). As a result, poorer performance on the test set might be expected. In addition, a sample of the confusion matrices (Table 2) for each of the methods indicates that many of the algorithms do reasonably well in classifying bent-doubles, with few bents classified as non-bents (considered desirable by the astronomers). However, some of the algorithms do not perform well in the classification of non-bents. For example, using Adaboost, with 10 trees and gain ratio as the splitting criterion, only 84 of the 198 non-bents are identified correctly. This is partly because the original training set has far fewer non-bents than bents. In addition, the current implementation of the splitting criterion does not treat misclassified non-bents different from misclassified bents in order to handle the unbalanced data set. We plan to incorporate this in a future implementation of our software and we expect that this will improve the results.

There are several other observations we can make from the results in Table 1. No splitting criterion gives consistently better accuracy for the different methods. Gain ratio is better for the histogram-based

| Method | Gini | Gain Ratio | Info Gain |
|---|---|---|---|
| Single tree | 32.41 | 42.76 | 30.00 |
| Histogram-based tree | 30.69 | 34.14 | 29.31 |
| Histogram-based ASPEN 10 trees | 33.17 | 27.97 | 32.93 |
| | (0.597) | (0.854) | (0.565) |
| Histogram-based ASPEN 20 trees | 32.24 | 26.55 | 32.28 |
| | (0.465) | (0.465) | (0.507) |
| Adaboost 10 trees | 34.83 | 43.10 | 42.41 |
| Adaboost 20 trees | 34.83 | 42.07 | 42.76 |
| Bagging 10 trees | 34.72 | 33.79 | 36.65 |
| | (0.974) | (1.087) | (1.61) |
| Bagging 20 trees | 32.38 | 32.24 | 33.65 |
| | (0.486) | (0.478) | (0.681) |
| ArcX4 10 trees | 41.38 | 42.41 | 38.28 |
| ArcX4 20 trees | 34.82 | 39.66 | 38.62 |

Table 1: Test error rates (std error) using different classification methods

| Method | Gini | Gain Ratio | Info Gain |
|---|---|---|---|
| Single tree | $\begin{bmatrix}74 & 18\\ 76 & 122\end{bmatrix}$ | $\begin{bmatrix}82 & 10\\ 114 & 84\end{bmatrix}$ | $\begin{bmatrix}72 & 20\\ 67 & 131\end{bmatrix}$ |
| Histogram-based tree | $\begin{bmatrix}71 & 21\\ 68 & 130\end{bmatrix}$ | $\begin{bmatrix}80 & 12\\ 87 & 111\end{bmatrix}$ | $\begin{bmatrix}71 & 21\\ 64 & 134\end{bmatrix}$ |
| Histogram-based ASPEN 10 trees | $\begin{bmatrix}65 & 27\\ 76 & 122\end{bmatrix}$ | $\begin{bmatrix}74 & 18\\ 53 & 145\end{bmatrix}$ | $\begin{bmatrix}74 & 18\\ 72 & 126\end{bmatrix}$ |
| Histogram-based ASPEN 20 trees | $\begin{bmatrix}72 & 20\\ 70 & 128\end{bmatrix}$ | $\begin{bmatrix}77 & 15\\ 60 & 138\end{bmatrix}$ | $\begin{bmatrix}71 & 21\\ 69 & 129\end{bmatrix}$ |
| Adaboost 10 trees | $\begin{bmatrix}80 & 12\\ 89 & 109\end{bmatrix}$ | $\begin{bmatrix}81 & 11\\ 114 & 84\end{bmatrix}$ | $\begin{bmatrix}78 & 14\\ 109 & 89\end{bmatrix}$ |
| Adaboost 20 trees | $\begin{bmatrix}80 & 12\\ 89 & 109\end{bmatrix}$ | $\begin{bmatrix}80 & 12\\ 110 & 88\end{bmatrix}$ | $\begin{bmatrix}77 & 15\\ 109 & 89\end{bmatrix}$ |
| Bagging 10 trees | $\begin{bmatrix}77 & 15\\ 82 & 116\end{bmatrix}$ | $\begin{bmatrix}75 & 17\\ 63 & 135\end{bmatrix}$ | $\begin{bmatrix}72 & 20\\ 83 & 115\end{bmatrix}$ |
| Bagging 20 trees | $\begin{bmatrix}75 & 17\\ 75 & 123\end{bmatrix}$ | $\begin{bmatrix}74 & 18\\ 76 & 122\end{bmatrix}$ | $\begin{bmatrix}78 & 14\\ 89 & 109\end{bmatrix}$ |
| ArcX4 10 trees | $\begin{bmatrix}80 & 12\\ 108 & 90\end{bmatrix}$ | $\begin{bmatrix}79 & 13\\ 110 & 88\end{bmatrix}$ | $\begin{bmatrix}80 & 12\\ 99 & 99\end{bmatrix}$ |
| ArcX4 20 trees | $\begin{bmatrix}81 & 11\\ 90 & 108\end{bmatrix}$ | $\begin{bmatrix}79 & 13\\ 102 & 96\end{bmatrix}$ | $\begin{bmatrix}79 & 13\\ 99 & 99\end{bmatrix}$ |

Table 2: Typical confusion matrices (bent, non-bent) for the different classification methods, corresponding to the results in Table 1. The two columns of the top row list the number of bents identified as bent and non-bent respectively. The two columns of the bottom row list the number of non-bents identified as bent and non-bent respectively.

ASPEN algorithm, but worse for a single tree and a histogram-based single tree. The least test error (26.55) was obtained with histogram-based ASPEN, 10 trees, using gain ratio for splitting. The best result with a single tree was 30.00, and the histogram-based single tree, though approximate, improved the results from the single tree for all three splitting criteria.

Next, we present the result of 10 runs of 10-fold cross-validation for the different algorithms using the original training set of 195 instances, plus the new validated set of 290 instances. The cross-validation error (and standard error) are reported in Table 3. Since the data set is now larger, with 485 instances, we also report the performance of the sampling-based ASPEN algorithm. In addition, we include results with pessimistic-error pruning as it can improve the performance. We also restrict our study to ensembles of 10 trees.

From the results presented in Table 3, we make the following observations. For this dataset, with the given set of features and training examples, the sampling-based ASPEN approach gave the best results (17.08), followed by the histogram-based ASPEN (18.02) and Bagging (18.12). They all improved the performance over a single tree (19.71). Even a single histogram-based tree (18.81) improves over a single tree. In comparison, Adaboost and ArcX4 did not perform as well.

In terms of the compute time, the ASPEN approaches typically tended to be faster than the other ensemble methods. They involve fewer computations as the decisions at each node of the tree consider fewer candidate splitting points. More detailed studies of the relative timing of these algorithms are given in [12, 14].

# 5 Summary

In this paper, we compared the performance of several algorithms for creating decision tree ensembles, using a data set from astronomy as our test-bed application. Given the set of features and the training data, the algorithms based on the ASPEN approach performed better than traditional ensemble based techniques such as Adaboost and Bagging.

# 6 Acknowledgements

We acknowledge the contributions of Imola Fodor and Nu Ai Tang of the Sapphire project, David Littau, who worked on the histogram-based ensembles as a summer student, and the FIRST scientists Robert Becker, Michael Gregg, David Helfand, Sally Laurent-Muehleisen, and Richard White. We also appreciate the insightful comments of the first reviewer.

UCRL-JC-146721: This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

# References

[1] Bauer, E., and Kohavi, R. An empirical comparison of voting classification algorithms: Bagging boosting and variants. *Machine Learning 36*, 1/2 (1999), 105–139.

[2] Becker, R. H., White, R., and Helfand, D. The FIRST survey: Faint images of the radio sky at twenty-cm. *Astrophysical Journal 450* (1995), 559.

[3] Breiman, L. Bagging predictors. *Machine Learning 26*, 2 (1996), 123–140.

[4] Breiman, L. Pasting bites together for prediction in large data sets and on-line. Tech. rep., Statistics Department, University of California, Berkeley, 1996. ftp.stat.berkeley.edu/-pub/users/breiman/pastebite.ps.Z.

[5] Breiman, L. Arcing classifiers. *Annals of Statistics 26* (1998), 801–824.

[6] Breiman, L. Random forests - random features. Tech. Rep. Technical Report 567, Statistics Department, University of California, Berkeley, 1999.

[7] Dietterich, T. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems* (2000), Springer Verlag, pp. 1–15.

[8] Dietterich, T. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning 40*, 2 (2000), 139–158.

[9] Fodor, I. K., Cantú-Paz, E., Kamath, C., and Tang, N. Finding bent-double radio galaxies: A case study in data mining. In *Interface : Computer Science and Statistics* (April 2000), vol. 33. See http://www.llnl.gov/casc/sapphire/pubs.html.

[10] Freund, Y., and Schapire, R. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference* (1996), pp. 148–156.

| Method | Gini | | Gain Ratio | | Info Gain | |
|---|---|---|---|---|---|---|
| | No Pruning | Pruning | No Pruning | Pruning | No Pruning | Pruning |
| Single tree | 22.79 (0.31) | 19.77 (0.18) | 22.62 (0.27) | 19.83 (0.15) | 22.77 (0.39) | 19.71 (0.41) |
| Histogram-based single tree | 21.73 (0.34) | 20.46 (0.29) | 20.85 (0.39) | 18.81 (0.17) | 22.56 (0.32) | 20.96 (0.39) |
| Histogram-based 10 trees | 18.69 (0.28) | 18.27 (0.30) | 18.10 (0.16) | 18.02 (0.22) | 18.22 (0.18) | 18.42 (0.34) |
| Sampling-based 10 trees (20%) | 20.40 (0.37) | 17.54 (0.27) | 18.62 (0.35) | 17.08 (0.17) | 18.73 (0.40) | 17.67 (0.29) |
| Sampling-based 10 trees (50%) | 18.21 (0.23) | 17.31 (0.17) | 17.83 (0.20) | 17.48 (0.15) | 17.79 (0.19) | 17.58 (0.18) |
| Adaboost 10 trees | 21.87 (0.42) | 20.40 (0.45) | 22.37 (0.53) | 22.56 (0.47) | 20.50 (0.45) | 20.75 (0.43) |
| Bagging 10 trees | 19.40 (0.28) | 18.35 (0.34) | 18.98 (0.34) | 18.12 (0.26) | 18.98 (0.36) | 18.52 (0.35) |
| ArcX4 10 trees | 20.48 (0.39) | 20.12 (0.20) | 21.67 (0.35) | 22.48 (0.41) | 21.06 (0.22) | 19.77 (0.37) |

Table 3: Cross-validation error (std error) using different classification methods, using 485 instances


[11] Hall, L., Bowyer, K., Kegelmeyer, W., Moore, T., and Chao, C. Distributed learning on very large data sets. In *Workshop on Distributed and Parallel Knowledge Discover, in conjunction with KDD2000* (2000).

[12] Kamath, C., and Cantú-Paz, E. Creating ensembles of decision trees through sampling. In *Proceedings, 33-rd Symposium on the Interface of Computing Science and Statistics* (June 2001).

[13] Kamath, C., Cantú-Paz, E., Fodor, I., and Tang, N. Searching for bent-double galaxies in the first survey. In *Data Mining for Scientific and Engineering Applications*, R. Grossman, C. Kamath, W. Kegelmeyer, V. Kumar, and R. Namburu, Eds. Kluwer, Boston, MA, 2001, pp. 95–114.

[14] Kamath, C., Cantú-Paz, E., and Littau, D. Approximate splitting for ensembles of trees using histograms. In *Proceedings, Second SIAM International Conference on Data Mining* (April, 2002). To appear.

[15] Murthy, K. V. S. *On Growing Better Decision Trees from Data.* PhD thesis, Johns Hopkins University, 1997.

[16] Quinlan, R. Simplifying decision trees. *Intl. J. Man-Machine Studies 27* (1987), 221–234.

[17] Sapphire: Large-scale data mining and pattern recognition. http://www.llnl.gov/casc/sapphire.

[18] White, R. L., Becker, R., Helfand, D., and Gregg, M. A catalog of 1.4 GHz radio sources from the FIRST survey. *Astrophysical Journal 475* (1997), 479.