Combining evolutionary algorithms with oblique decision trees to detect bent-double galaxies *

Erick Cantú-Paz and Chandrika Kamath

Center for Applied Scientific Computing, Lawrence Livermore National Laboratory,
P.O. Box 808, L-561, Livermore, CA 94551

ABSTRACT

Decision trees have long been popular in classification as they use simple and easy-to-understand tests at each node. Most variants of decision trees test a single attribute at a node, leading to axis-parallel trees, where the test results in a hyperplane which is parallel to one of the dimensions in the attribute space. These trees can be rather large and inaccurate in cases where the concept to be learned is best approximated by oblique hyperplanes. In such cases, it may be more appropriate to use an oblique decision tree, where the decision at each node is a linear combination of the attributes. Oblique decision trees have not gained wide popularity in part due to the complexity of constructing good oblique splits and the tendency of existing splitting algorithms to get stuck in local minima. Several alternatives have been proposed to handle these problems including randomization in conjunction with deterministic hill-climbing and the use of simulated annealing. In this paper, we use evolutionary algorithms (EAs) to determine the split. EAs are well suited for this problem because of their global search properties, their tolerance to noisy fitness evaluations, and their scalability to large dimensional search spaces. We demonstrate our technique on a synthetic data set, and then we apply it to a practical problem from astronomy, namely, the classification of galaxies with a bent-double morphology. In addition, we describe our experiences with several split evaluation criteria. Our results suggest that, in some cases, the evolutionary approach is faster and more accurate than existing oblique decision tree algorithms. However, for our astronomical data, the accuracy is not significantly different than the axis-parallel trees.

Keywords: classification, decision trees, oblique decision trees, evolutionary algorithms, data mining

1. INTRODUCTION

Data mining is a process concerned with uncovering patterns, associations, anomalies and statistically significant structures in data.[1,2] It consists of two main steps, data pre-processing, during which relevant features or attributes are extracted from the data, and pattern recognition, in which a pattern in the data is recognized using these features. An important category of pattern recognition algorithms is classification. In these algorithms, a set of known or labeled data instances, referred to as the training set, is used by the algorithm to create a model that predicts the label of unseen instances.

Decision trees (DTs) are an important category of classification algorithms in light of their ease of construction and interpretation. Most decision tree algorithms create tests at each node that involve a single feature or attribute. These trees are referred to as axis-parallel trees because the tests can be considered as hyperplanes that are parallel to one of the axes in the attribute space. As the tests at each node are very simple, it is easy for the domain expert to interpret the trees. However, in some cases, where the data is more accurately partitioned using hyperplanes that are not axis-parallel, this simple univariate approach may result in complicated and inaccurate trees. A possible solution to this problem is the use of oblique decision trees, where the decision at each node is a linear combination of the features. The resulting trees, while smaller and more accurate, may require greater computational effort and be harder to interpret.

In this paper, we illustrate the use of evolutionary algorithms to induce oblique decision trees. In our approach, we use evolutionary algorithms to solve the optimization problem at each node of the oblique decision tree. Using an artificial data set, we first show that in some domains, oblique decision trees are a better alternative to axis-parallel

*Accepted for publications in the Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation III, at the International Society for Optical Engineering (SPIE) Annual Meeting, 2000

Further author information: E.C.P: E-mail: cantupaz1@llnl.gov; C.K.: E-mail: kamath2@llnl.gov
(Send correspondence to C.K.)

trees. In addition, we show that oblique trees created using evolutionary algorithms may require less computational time and be more robust with increasing dimensionality of the problem than traditional oblique DTs. Using a real data set from astronomy, we discuss our experiences in applying these decision trees. Our results suggest that the evolutionary approach is faster and more accurate than the existing oblique DTs in this data set, but the accuracy is not significantly better than with axis-parallel trees. We also discuss the effect of varying the criterion used to evaluate the split on this data set.

The paper is organized as follows: the next section introduces oblique decision trees and the split criteria that can be used to evaluate the "goodness" of a decision at a node of the tree. Section 3 describes the use of two evolutionary algorithms in calculating the decision at a node and briefly summarizes our prior experiences with artificial data sets. In Section 4 we describe the problem of detection of radio-emitting galaxies with a bent-double morphology, and in Section 5 we present our results from applying various decision tree algorithms and split criteria to this problem. We discuss the results in Section 6, and conclude the paper in Section 7 with a summary.

2. OBLIQUE DECISION TREES

The input to the decision tree algorithm are data instances of the form $(x_1, x_2, \ldots, x_d, c_j)$, where the x_i are the real-valued features of the instance and c_j is the discrete class label assigned to the instance. At each node of the tree, an axis-parallel algorithm evaluates a test of the form $x_i > k$. The task is to find the values of i and k that best partition the data instances at each node of the tree. In contrast, in an oblique decision tree, a more general test of the form

$$\sum_{i=1}^{d} a_i x_i + a_{d+1} > 0 \tag{1}$$

is considered at each node. The task in this case is to find the best combination of the a_i that partitions the data instances at a node. This is essentially a search in a $(d+1)$ dimensional space. As a result, the task of the algorithm is much harder than in the axis parallel case. This problem is an NP-complete problem,[3] and therefore, oblique decision tree classifiers typically use a greedy search to determine the coefficients. Regardless of the way in which the split or the partition is determined, the building of the tree starts with the entire training set at the root node. The best partition for the root is determined, the data is split into subsets based on this test, and the algorithm is applied recursively to each subset.

The first algorithm to induce oblique decision trees was CART-LC (CART with Linear Combinations), described in the classic text by Breiman, Friedman, Olshen, and Stone.[4] They used a deterministic approach to iteratively find locally optimal values for each of the coefficients a_i. However, this technique could potentially get stuck in a local optimum. To overcome this problem, Murthy, Kasif, and Salzberg[5] introduced the OC1 (Oblique Classifier 1) algorithm, which used an ad-hoc combination of hill-climbing and randomization. As in CART-LC, their approach used hill-climbing to find the locally optimum solution for each coefficient at a time. However, they coupled this with randomization to avoid getting trapped in a local optimum. They considered two forms of randomization: multiple random restarts and random perturbations of the hyperplane determined in the hill-climbing process. Their results suggested that in some domains, OC1 outperformed CART-LC.

Regardless of the techniques used to determine the hyperplane, all decision tree algorithms need a heuristic measure to evaluate the goodness of a proposed split. Each such measure defines a different optimization problem. Depending on whether the measure evaluates the goodness or badness of a split, it can be either maximized or minimized. Let T be the set of n examples at a node that belong to one of k classes, and T_L and T_R be the two non-overlapping subsets that result from the split (that is, the left and right subsets). Let L_j and R_j be the number of instances of class j on the left and the right, respectively. Some of the commonly used measures, and their advantages and disadvantages are[6]:

- **Gini index:** This criterion is based on finding the split that most reduces the node impurity, where the impurity is defined as follows:

$$L_{Gini} = 1.0 - \sum_{i=1}^{k} (L_i/|T_L|)^2, \qquad R_{Gini} = 1.0 - \sum_{i=1}^{k} (R_i/|T_R|)^2 \tag{2}$$

$$\text{Impurity} = (|T_L| * L_{Gini} + |T_R| * R_{Gini})/n$$

where $|T_L|$ and $|T_R|$ are the number of examples, and L_{Gini} and R_{Gini} are the Gini indices on the left and right side of the split, respectively. This criterion can have problems when there are a large number of classes.

- **Twoing rule:** In this case, the "goodness" of the split is evaluated as follows:

$$\text{Twoing value} = (|T_L|/n) * (|T_R|/n) * \left(\sum_{i=1}^{k} |L_i/|T_L| - R_i/|T_R|| \right)^2 \tag{3}$$

Twoing may be a more desirable split criterion when there are a large number of classes. Comparisons between twoing and Gini indicate only slight differences between the two, though Breiman et al.[4] prefer Gini as the splits produced by it appear to be better in terms of producing pure descendant nodes.

- **Information gain:** The information gain associated with a feature is the expected reduction in entropy caused by partitioning the examples according to the feature. Here the entropy characterizes the (im)purity of an arbitrary collection of examples. For example, the entropy prior to the split in our example would be:

$$\text{Entropy}(T) = \sum_{i=1}^{k} -p_i \log_2 p_i, \tag{4}$$

where p_i is the proportion of T belonging to class i. The information gain of a split $S = \{T_L, T_R\}$ relative to T is then given by

$$\text{Gain}(T,S) = \text{Entropy}(T) - \frac{|T_L|}{|T|} * \text{Entropy}(T_L) - \frac{|T_R|}{|T|} * \text{Entropy}(T_R) \tag{5}$$

where T_L and T_R are the subsets of T that correspond to the left and right branches, respectively. This criterion tends to favor features with many values over those with few values.

3. EVOLUTIONARY OBLIQUE DECISION TREES

At the heart of all decision tree algorithms there is an optimization task, namely, the optimization of the measure used for determining the quality of a split or partition. It is therefore natural to consider applying evolutionary algorithms to solve this optimization problem. In an earlier paper,[7] we considered the use of two such algorithms to find the best oblique partition for a tree:

- Oblique-ES: In this case, we used a (1+1) evolution strategy with self adaptive mutations. The candidate hyperplane is represented as a vector of $(d+1)$ real numbers that represent the coefficients of the hyperplane. This is initially set to the best axis-parallel hyperplane. For each hyperplane coefficient, there is a corresponding mutation coefficient $\sigma_1, \ldots, \sigma_{d+1}$, which are initially set to 1. At each iteration, the mutation coefficients are updated and a new hyperplane is obtained according to the rule

$$\begin{aligned} \nu &= N(0,1) \\ \sigma_i^{t+1} &= \sigma_i^t \exp(\tau' \nu + \tau N(0,1)) \\ a_i^{t+1} &= a_i^t + \sigma_i^{t+1} N(0,1), \end{aligned} \tag{6}$$

where $N(0,1)$ indicates a realization of a unit normal variate, $\tau = (\sqrt{2\sqrt{d}})^{-1}$, and $\tau' = (\sqrt{2d})^{-1}$. The Oblique-ES algorithm was stopped after 1000 iterations.

- Oblique-GA: In this algorithm, we use a simple generational GA with real valued alleles that represent the coefficients of the hyperplane. For our experiments, we used pairwise tournament selection without replacement, uniform crossover with probability 1.0, and no mutation. The population size was set to $20\sqrt{d}$ based on a recent theory that suggests the size of the population needed to reach a solution of a particular quality.[8] The initial population consisted of 90% individuals with random coefficients in the range [-200,200], and 10% individuals that are identical to the best axis parallel solution. The GA was stopped after 25 generations. The best hyperplane found by the GA was compared with the original axis parallel, and the best one was used to split the data.

Algorithm	Parameter	LS10	LS20	LS50
OC1-AP	Error	27.0 (0.47)	35.4 (0.25)	41.4 (0.31)
	Leaves	86.7 (5.2)	71.5 (9.1)	58.0 (6.5)
	Time	1.6 (0.0)	3.5 (0.03)	11.7 (0.18)
OC1	Error	2.9 (0.12)	11.5 (0.34)	27.5 (0.41)
	Leaves	5.3 (0.69)	5.9 (0.85)	10.0 (1.13)
	Time	170.9 (3.8)	391.5 (5.2)	608.7 (10.4)
OC1-CART	Error	4.0 (0.47)	12.7 (0.6)	33.7 (0.31)
	Leaves	5.9 (1.1)	9.3 (1.1)	25.0 (5.6)
	Time	16.8 (0.4)	54.9 (1.1)	113.9 (1.1)
Oblique-ES	Error	6.3 (0.25)	13.0 (0.31)	21.5 (0.5)
	Leaves	9.9 (0.9)	14.4 (1.7)	16.3 (2.9)
	Time	29.8 (0.75)	65.1 (1.0)	163.9 (4.7)
Oblique-GA	Error	4.6 (0.18)	8.0 (0.22)	14.8 (0.31)
	Leaves	8.8 (1.2)	9.8 (1.8)	9.5 (1.7)
	Time	36.3 (3.8)	101.5 (1.5)	333.3 (7.0)

Table 1. Comparison of different algorithms on the linearly-separable data sets.

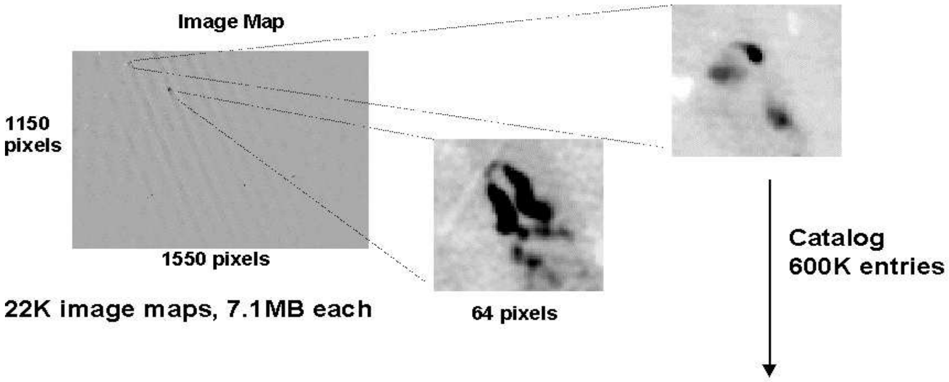
In a previous study, we found that the parameters listed above for each algorithm perform well on several classification problems,[7] but we recognize that we can get better performance if we fine tune the parameters to the particular problem instance (also, we can expect that these parameters will not work well on some problems). However, we decided not to tune the parameters of the EAs, because we do not adjust the default parameters of the other algorithms.

We compared the performance of our EA-based decision trees with other three algorithms: OC1, OC1 restricted to standard axis-parallel splits (OC1-AP), and the version of CART-LC that is included in the OC1 package (OC1-CART).[7] In each case, we used the Twoing rule (Equation 3) to measure the quality of a split. We summarize the results on three data sets that were generated artificially to have oblique partitions. The data sets have 2000 instances divided into two classes. Each instance has d attributes with values uniformly distributed in [0,1]. The data is separable by the hyperplane $x_1 + \cdots + x_{d/2} < x_{d/2+1} + \cdots + x_d$, where $d \in \{10, 20, 50\}$. The data sets are labeled LS10, LS20, and LS50 to reflect their dimensionality.

To estimate the performance of our algorithms, we performed ten, five-fold cross-validation experiments on each data set, which is the procedure followed by Murthy et al. to evaluate OC1. The results are summarized in Table 1. The table shows the mean error rate, the average number of leaves of the trees found (after pruning), and the average time (in seconds). The numbers in parenthesis are the standard errors for each result. The experiments were executed on a 500 MHz Pentium III PC with 128 Mb of RAM running NT 4.0. The programs were compiled with the egcs compiler version 2.91 using `-O` optimizations.

As expected, OC1-AP consistently found the least accurate and largest trees. Of course, it was the fastest algorithm, but its error rate is too high to consider it competitive (consider that random guessing would result in a 50% accuracy and the error rate of OC1-AP on LS50 is 41%). OC1 produces the most accurate trees for LS10, but as the number of dimensions increases its performance seems to drop below the EA-based inducers, and OC1-CART does slightly worse. OC1-GA maintains the highest accuracy, but its execution time seems to increase faster than OC1-ES. In any case, both of the EA inducers are faster than OC1 (approximately between 2x and 6x), and appear to find more accurate classifiers than the other algorithms as the number of dimensions increase. This has important practical consequences, because the data instances may be described by numerous attributes, as is the case with the data set described in the next section.

The size of the trees found by OC1, OC1-CART, and OC1-ES increases with the number of dimensions, but those of OC1-GA seem to remain of a constant size. Consider that the data can be separated by a single hyperplane, and therefore the ideal tree for this domain has only two leaves, but all the algorithms found much larger trees.



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. A FIRST image map with two bent-double galaxies and the catalog entries

4. DETECTION OF BENT-DOUBLES IN THE FIRST DATA SET

To explore the effectiveness of the EA-based decision trees on a real problem, we used them on a practical problem in astronomy, namely, the detection of bent-double radio galaxies in the FIRST data set. The Faint Images of the Radio Sky at Twenty-cm (FIRST)[9] is an astronomical survey whose goal is to produce the radio equivalent of the Palomar Observatory Sky Survey. Using the Very Large Array at the National Radio Astronomy Observatory, FIRST is scheduled to cover more than 10,000 square degrees of the northern and southern galactic caps, to a flux density limit of 1.0 mJy (milli-Jansky). At present, with the data from the 1993 through 1998 observations, FIRST has covered about 6,000 square degrees, producing more than 20,000 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.

Radio sources exhibit a wide range of morphological types that provide clues to the source class, emission mechanism, and properties of the surrounding medium. The FIRST astronomers are particularly interested in radio sources 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, is also becoming increasingly infeasible as the survey grows in size. Our goal is to automate this process of classifying galaxies.

Raw and post-processed data from the FIRST survey are accessible on the FIRST website (http://sundog.stsci.edu). There are two forms of data available: 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. Each map covers approximately 0.45 square degrees area of the sky, and has pixels which are 1.8 arc seconds wide. The FIRST catalog[10] is obtained by processing an image map 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).

One of the first steps in the semi-automatic detection of bent-doubles using data mining techniques was the extraction of relevant features from the FIRST data. Our conversations with the astronomers indicated that they believed the catalog to be a good approximation to all but the most complex of radio sources. As a result, they felt that we could extract most, if not all, of the relevant features from the catalog itself, without having to use the images as well.

We used the following process to extract features from the FIRST catalog:

- We first grouped the entries in the catalog to identify those which contributed to a radio source. Based on the information from FIRST astronomers, we considered any Gaussians within 0.96 arc-minutes of each other to constitute a single radio source or radio-emitting galaxy.
- We then separated the radio sources based on the number of Gaussians that formed the source. For the 1999 catalog, including observations from 1993 through 1998, the number of radio sources as a function of the number of catalog entries or Gaussians used to approximate them are as follows:

Number Catalog Entries	**Number Radio Sources**
1	392360
2	50498
3	11439
4+	4764

- Next, we focused on sources with two or three catalog entries. We expected that sources approximated by one Gaussian were unlikely to be bent doubles. Also, sources that were approximated by more than three Gaussians were complex enough to be of interest to the astronomers regardless of whether they were bent doubles or not.
- As the number of features extracted for a radio source was dependent on the number of Gaussians or catalog entries in the source, we decided to create separate decision trees for two- and three-catalog entry sources. This unfortunately 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. Note that the small size of the training set is the result of the manual detection of the bent doubles by astronomers, which is a very tedious task.
- We next extracted relevant features from the catalog for both the two and three-entry sources. A complete list of the features is given in an earlier paper on our work on FIRST.[11] In addition to the features from the catalog, we included derived features such as the relative distances between the centers of the Gaussians, and angles in the triangle formed by the centers of the Gaussians in a three-entry source.
- Once we had a preliminary set of features, we performed experiments with C5 (Rulequest, Inc.), an axis-parallel tree inducer, to understand and refine the set of features extracted.[11] We found that in the three-entry case the error was approximately 9%, but the error for the two-entry case was closer to 20%. We suspect that there are several reasons for the poor performance for the two-entry case such as a small training set, information in the catalog that is not representative of the images, and possible inclusion of non-relevant features while missing the relevant ones. We are investigating this issue further. For this paper, we will focus on the three-entry radio sources.

The next section describes our experiments with the FIRST data set using existing oblique decision tree inducers, as well as our extensions with evolutionary algorithms.

5. EXPERIMENTAL RESULTS WITH THE FIRST DATA

The training set that we used in the experiments has 195 examples, and each example is described by 103 numeric features. In the FIRST domain, there are two possible classes: an instance describes either a bent or a non-bent galaxy. We evaluated the performance of the same five algorithms described above using ten ten-fold cross-validation experiments. For each algorithm, we tried three different impurity measures: twoing, Gini index, and the information gain, which are described in Section 2. The results are summarized in Table 2. The table shows the mean error rate, the average number of leaves of the trees found (after pruning), and the average time (in seconds). The numbers in parenthesis are the standard errors for each result.

The results show that the EA-based oblique DT inducers have better accuracy than the other oblique algorithms. The Oblique-ES and the Oblique-GA consistently give better solutions than OC1 and OC1-CART. The Oblique-ES trees have approximately 30–40% fewer classification errors than the existing oblique DTs, and the GA has approximately 15–35% fewer errors. In all cases, the evolutionary algorithms are faster than the original OC1, but CART is the fastest overall oblique DT. However, the accuracy of the best oblique trees is not significantly different than the axis-parallel trees, and it seems that AP trees are a good choice in the FIRST domain.

Algorithm	Parameter	Twoing	Gini	Info Gain
	Error	8.62 (0.58)	8.14 (0.54)	9.17 (0.67)
OC1-AP	Leaves	3.63 (0.41)	3.68 (0.42)	4.3 (0.37)
	Time	2.97 (0.07)	2.84 (0.06)	4.72 (0.09)
	Error	12.76 (0.61)	13.16 (0.39)	14.29 (0.59)
OC1	Leaves	2.9 (0.13)	2.86 (0.12)	2.96 (0.11)
	Time	44.73 (1.0)	252 (14.5)	116 (2.1)
	Error	13.13 (0.56)	12.51 (0.68)	13.12 (0.64)
OC1-CART	Leaves	2.61 (0.16)	2.82 (0.14)	2.69 (0.13)
	Time	10.37 (0.27)	9.22 (0.22)	26.64 (1.17)
	Error	8.69 (0.58)	8.03 (0.59)	8.83 (0.59)
Oblique-ES	Leaves	3.59 (0.41)	3.9 (0.40)	4.24 (0.36)
	Time	34.96 (0.92)	34.5 (0.86)	36.21 (0.63)
	Error	10.76 (0.65)	8.65 (0.64)	11.96 (0.71)
Oblique-GA	Leaves	3.54 (0.14)	3.54 (0.15)	3.71 (0.18)
	Time	162 (2.9)	112 (3.2)	170 (3.4)

Table 2. Comparison of different algorithms on the FIRST data set.

The table also shows that the solution accuracy improves slightly when the Gini index is used to evaluate the splits, except for OC1 where the Twoing rule gives the best results. The differences in accuracy are not significant, however, except in the case of the GA. The tree sizes also seem unaffected by the split evaluation criteria, but in some cases there are large differences in the training time. The Gini index seems to need the shortest training times for all algorithms except OC1, but some of the differences are not significant.

The differences in training time may be caused by several factors, which we are currently investigating. We suspect that some combinations of algorithms and split impurity measures result in not-so-good splits near the root of the tree in this particular domain. If this were the case, the algorithm would continue to split the data until it finds relatively pure partitions, which requires additional hyperplane evaluations and longer training times. If good splits are found near the root, the algorithm would stop earlier, requiring a shorter training time. Note that the slowest algorithms are also the least accurate, which is consistent with our hypothesis.

6. DISCUSSION OF RESULTS

In this Section, we attempt to explain the performance of the different decision tree algorithms on the FIRST data set. In particular, we would like to better understand the reason why the mean error rate with oblique decision trees is no better than the axis-parallel tree. While it is well known that the performance of classification algorithms can vary depending on the problem, little is known a priori about which algorithm will work best on a problem. We hope to shed some light on this, at least for our problem of the detection of bent-double radio-emitting galaxies.

Figure 2 has a typical tree generated by the axis-parallel algorithm. This output lists the feature selected at each node, as well as the value it is compared against. The number after the colon indicates that the node in question is a leaf node, and the number is the class assigned to the leaf (5 denotes a bent double, while 1 denotes a non-bent double). The numbers in parenthesis at each leaf node, (a/b), indicate that of the total samples a at that node, $(a-b)$ samples belonged to the class assigned to the leaf node, while b samples were of the class not assigned to the leaf node. The tree has a depth of four, with six leaves.

Our conversations with the FIRST astronomers indicated that the features selected by the tree, such as the relative distances and angles, were very relevant to the identification of bent-doubles. So, from a domain point of view, the axis-parallel tree agreed with the astronomers' interpretation of how they would identify a bent-double. Note that the leaf nodes tend to be very pure, that is, they are composed of only one class. Also, a substantial percentage of the examples (90% = 158 out of 176) have been correctly classified by a tree of depth two. In addition, 81% of the examples (143 out of 176), can be classified as bent-doubles based solely on two features using axis parallel

```
rs3_core_angle > 170.4:
:...cec_ellipticity <= 2.116: 1 (13.0)
:   cec_ellipticity > 2.116: 5 (2.0)
rs3_core_angle <= 170.4:
:...pairac_rel_dist <= 9.423: 5 (143.0)
    pairac_rel_dist > 9.423:
    :...pairab_angle_geom <= 58.6: 5 (4.0/1.0)
        pairab_angle_geom > 58.6:
        :...cec_rms <= 0.137: 5 (5.0/2.0)
            cec_rms > 0.137: 1 (9.0)
```

Figure 2. An axis-parallel decision tree for the FIRST domain.

hyperplanes. All these observations seem to indicate that the classification of bent-doubles, based on the features we have extracted, is a problem very well suited for axis-parallel trees.

These observations may give us a possible approach to selecting a decision tree algorithm for a problem. Since axis-parallel trees take relatively little time to generate, even for large trees, we can easily generate one to understand how it partitions the features space. If it creates a very large tree, with few samples at each leaf, and leaves with low purity, an oblique classifier may be a better solution. In such cases, oblique classifiers based on evolutionary algorithms, such as the ones proposed in this paper, can be an attractive alternative to the standard oblique classifier.

7. SUMMARY AND CONCLUSIONS

In this paper, we substitute the heuristic greedy search used by existing oblique DT inducers with two evolutionary algorithms: a (1+1) evolution strategy and a simple GA. Our previous experience with these algorithms suggest that they are capable of finding accurate oblique trees at a competitive cost, and that they scale up better than existing methods to the dimensionality of the data. These reasons encouraged us to try these algorithms on the problem of identifying bent-double galaxies in the FIRST data set.

Our results with the FIRST data are consistent with our previous findings: the evolutionary algorithms found trees that are more accurate than those found by OC1 and CART-LC. However, the best oblique trees are not significantly more accurate than the axis-parallel trees, and it appears that AP trees are sufficient to identify bent-double galaxies with very high accuracy.

There are multiple opportunities to expand our work. In particular, we will continue the study of scalability using larger data sets, and incorporate other optimization methods, both evolutionary such as (μ, λ)-ES and traditional methods. Also, to improve the performance we will explore the option of combining the EAs with the hill-climbing algorithms used in existing DT inducers.

ACKNOWLEDGMENTS

We gratefully acknowledge our FIRST collaborators Robert H. Becker, Michael D. Gregg, David J. Helfand, Sally A. Laurent-Muehleisen, and Richard L. White for their technical interest and support of this work. We would also like to thank Imola K. Fodor and Nu Ai Tang for their computational help.

UCRL-JC-138979. 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. U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD Process for Extracting Useful Knowledge from Volumes of Data," *Communications of the ACM Special Issue on Data Mining* **39**, pp. 27–34, 1996.
2. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge, Mass., 1996.
3. D. Heath, S. Kasif, and S. Salzberg, "Induction of Oblique Decision Trees," in *Proceedings of the 13-th Joint Conference on Artificial Intelligence*, pp. 1002–1007, Morgan Kaufmann, (San Mateo, California), 1993.

4. L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Chapman and Hall/CRC Press, Boca Raton, Florida, 1984.
5. S. Murthy, S. Kasif, and S. Salzberg, "A System for Induction of Oblique Decision Trees," *Journal of Artificial Intelligence Research* **2**(1), pp. 1–32, 1994.
6. K. V. S. Murthy, *On Growing Better Decision Trees from Data.* PhD thesis, Johns Hopkins University, 1997.
7. E. Cantú-Paz and C. Kamath, "Using Evolutionary Algorithms to Induce Oblique Decision Trees," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2000.
8. G. Harik, E. Cantú-Paz, D. Goldberg, and B. Miller, "The Gambler's Ruin Problem, Genetic Algorithms, and the Sizing of Populations," *Evolutionary Computation* **7**(3), pp. 231–254, 1999.
9. R. Becker, R. White, and D. J. Helfand, "The FIRST Survey: Faint Images of the Radio Sky at Twenty-cm," *Astrophysical Journal* **450**, p. 559, 1995.
10. R. White, R. Becker, D. Helfand, and M. Gregg, "A Catalog of 1.4GHz Radio Sources from the FIRST Survey," *Astrophysical Journal* **475**, p. 479, 1997.
11. I. Fodor, E. Cantú-Paz, C. Kamath, and T. N., "Finding Bent-Double Radio Galaxies: A Case Study in Data Mining," in *Proceedings of the Interface: Computer Science and Statistics Symposium*, vol. 33, 2000.