USING MACHINE LEARNING TECHNIQUES TO AUTOMATE SKY SURVEY CATALOG GENERATION

Usama M. Fayyad†, Nicholas Weir‡, J.C. Roden†, S.G. Djorgovski‡, and R.J. Doyle†

†Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

‡Department of Astronomy
California Institute of Technology
Pasadena, CA 91125

ABSTRACT

We describe the application of machine classification techniques to the development of an automated tool for the reduction of a large scientific data set. The 2nd Palomar Observatory Sky Survey provides comprehensive photographic coverage of the northern celestial hemisphere. The photographic plates are being digitized into images containing on the order of 10^7 galaxies and 10^8 stars. Since the size of this data set precludes manual analysis and classification of objects, our approach is to develop a software system which integrates independently developed techniques for image processing and data classification. Image processing routines are applied to identify and measure features of sky objects. Selected features are used to determine the classification of each object. GID3* and O-BTree, two inductive learning techniques, are used to automatically learn classification decision trees from examples. We describe the techniques used, the details of our specific application, and the initial encouraging results which indicate that our approach is well-suited to the problem. The benefits of the approach are increased data reduction throughput, consistency of classification, and the automated derivation of classifications rules that will form an objective, examinable basis for classifying sky objects. Furthermore, astronomers will be freed from the tedium of an intensely visual task to pursue more challenging analysis and interpretation problems given automatically catalogued data.

1. INTRODUCTION

In this paper we present an application of machine learning techniques to the automation of the task of cataloguing sky objects in digitized sky images. The Sky Image Cataloging and Analysis Tool (SkICAT) is being developed for use on the images resulting from the 2nd Palomar Observatory Sky Survey (POSS-II) conducted by the California Institute of Technology (Caltech). The photographic plates collected from the survey are being digitized at the Space Telescope Science Institute (STScI). This process will result in about 1788 digital images of roughly $23{,}000^2$ pixels each. Each image is expected to contain on the order 10^5 sky objects.

The first step in analyzing the results of a sky survey is to identify, measure, and catalog the detected objects in the image into their respective classes. Once the objects have been classified, further scientific analysis can proceed. For example, the resulting catalog may be used to test models of the formation of large-scale structure in the universe, probe galactic structure from star counts, perform automatic identifications of radio or infrared sources, and so forth. The task of reducing the images to catalog entries is a laborious time-consuming process. A manual approach to constructing the catalog implies that many scientists need to expend large amounts of time on a visually intensive task that may involve significant subjective judgment. The goal of our project is to automate the process, thus alleviating the burden of cataloguing objects from the scientist and providing a more objective methodology for reducing the data sets. Another goal of this work is to classify objects whose intensity (isophotal magnitude) is too faint for recognition by inspection, hence requiring an automated classification procedure. Faint objects constitute the majority of objects on any given plate. We plan to automate the classification of objects that are at least one magnitude fainter than objects classified in previous surveys using comparable photographic material.

The goals of this paper are:

1. to introduce the machine learning techniques used and emphasize their general applicability to other data reduction or diagnostic classification tasks, and
2. to give a general, high-level description of the current application domain.

We therefore do not provide the details of either the learning algorithms or the technical aspects of the domain. We aim to point out an instance where the learning algorithms proved to be useful and powerful tool in the automation of scientific data analysis.

2. MACHINE LEARNING BACKGROUND

One of the goals of Artificial Intelligence (AI) research is to provide mechanisms for emulating human decision-making and problem solving capabilities, using computer programs. The first AI attempts at such systems appeared as part of the technology known as "expert systems". However, serious difficulties arose that pointed out the difficulties in endowing a system with sufficient knowledge to execute a specific task. The first such difficulty is the "knowledge acquisition bottleneck" [Feig81] due to experts finding it difficult to express their knowledge, or explain their actions, in terms of concise situation-action rules. The second problem arises in a different situation: What if a task is not well-understood, even by the experts in that area? Many processes are not well-understood and thus even experts cannot predict the outputs for a given set of inputs. An example of this situation is manifested in previous experience with the automation of the reactive ion etching (RIE) process in semiconductor manufacturing [Chen90]. In such domains, abundant data are available from the experiments conducted, or items produced. However, models that relate how output variables are affected by changes in the controlling variables are not available. Experts strongly rely on familiarity with the data and on "intuitive" knowledge of such a domain. How would one go about constructing an expert system for such an application?

The machine learning approach to circumventing the aforementioned hurdles calls for extracting classification rules from data directly. Rather than require that a domain expert provide knowledge, the learning algorithm attempts to discover, or induce, rules that emulate expert decisions in different circumstances by observing examples of expert tasks. The basic approach is described in the next section.

Two other reasons exist for the need for a machine learning approach. The first is the growing number of large diagnostic and scientific databases. A database that stores instances of diagnostic tasks is typically accessed by keyword or condition lookup. As the size of the database grows, such an approach becomes ineffective since a query may easily return hundreds of matches making simple case-based usage impractical. For large scientific databases the problem is to search for and detect patterns of interest, or to perform pre-processing necessary for subsequent analysis. Sizes are now becoming too large for manual processing. Learning techniques can serve as effective tools for aiding in the analysis, reduction, and visualization of large scientific databases. Another motivation is the evolution of complex systems that have an error detection capability. Communication networks are an example. Faults are detectable by the network hardware. Several thousand faults may occur during a day. To debug such a network, a human would need to sift through large amounts of data in search of an explanation. An automated capability of deriving conditions under which certain faults occur may be of great help to the engineer in uncovering underlying problems in the hardware.

2.1. THE MACHINE LEARNING APPROACH

The machine learning approach prescribes inducing classifiers by automatically analyzing classified examples rather than interviewing domain experts. A training example consists of a description of a situation and the action performed by the expert in that situation. The situation is described in terms of a set of attributes or variables. An attribute may be *continuous* (numerical) or *discrete* (nominal). For example, a nominal attribute may be *shape* with values {*square, triangle, circle* }. Examples of continuous attribute are *pressure, area,* or *temperature.* The action associated with the situation, the *class* to which the example belongs, is a specification of one of a fixed set of pre-defined classes. The class of each training example is typically determined by a human expert during normal task execution. Example actions (classes) in a diagnostic setting may be *raise temperature, decrease pressure, accept batch,...* If the classes represent diagnostic actions, then the classification problem becomes equivalent to diagnosis. The goal of the learning program is to derive conditions, expressed in terms of the attributes, that are predictive of the classes. Such rules may then be used by an expert system to classify future examples. Of course, the quality of the rules depends on the validity of the conditions chosen to predict each action.

A training example is a vector of attribute values along with the class to which the example belongs. Assume there are m attributes $A_1,...,A_m$, p classes $C_1,...,C_p$. A training example is an $m+1$-tuple $\langle b_1,b_2,...,b_m ; C_j\rangle$, where each b_i is one of the values of the attribute A_i: $\{a_{i1},...,a_{ir_i}\}$, and C_j is one of the p classes. A rule for predicting some class C_j consists of a specification of the values of one or more attributes on the left hand side and a specification of the class on the right hand side.

For example, consider the simplified small example set shown in Table 1. It consists of six examples e-1 through e-6. There are two attributes: S and L. The attributes can take the values *low, normal,* and *high.*

example	S	L	class
e-1	normal	normal	*PH*
e-2	normal	high	*PL*
e-3	high	high	*PL*
e-4	high	low	*PH*
e-5	low	normal	*FRL*
e-6	low	high	*FRL*

Table 1: A Simple Training Set.

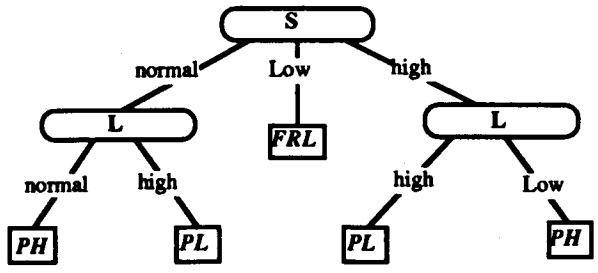

Figure 1: Decision Tree Generated by ID3 for Data Set of Table 1.

There are three classes: *FRH*, *PL*, and *PH*. A simple rule consistent with these examples may be:

IF (S = low) THEN class is *FRL*

Note that this is only an illustrative simplification. Typically, the number of examples of a meaningful training set is at least in the hundreds, while the number of attributes is usually in the tens.

2.2. INDUCING RULES FROM TRAINING EXAMPLES

Assume that there are m attributes as described above. Let each attribute A_i take on one of r_i values $\{ a_{i1},..., a_{ir_i} \}$. Assuming that on average an attribute takes on one of r values, there are $p(r+1)^m$ possible rules for predicting the p classes. It is computationally infeasible for a program to explore the space of all possible classification rules. In general, the problem of determining the minimal set of rules that cover a training set is NP-hard. Hence, there is no known computationally feasible (polynomial) algorithm for finding the solution. It is therefore likely that a heuristic solution to the problem is the only feasible one. A particularly efficient method for extracting rules from data is to generate a decision tree [Brei84, Quin86]. A decision tree consists of nodes that are tests on the attributes. The outgoing branches of a node correspond to all the possible outcomes of the test at the node. The examples at a node in the tree are thus *partitioned* along the branches and each child node gets its corresponding subset of examples. A popular algorithm for generating decision trees is Quinlan's ID3 [Quin86], now commercially available.

ID3 starts by placing all the training examples at the root node of the tree. An attribute is selected to partition the data. For each value of the attribute, a branch is created and the corresponding subset of examples that have the attribute value specified by the branch are moved to the newly created child node. The algorithm is applied recursively to each child node until either all examples at a node are of one class, or all the examples at that node have the same values for all the attributes. An example decision tree generated by ID3 for the sample data set given in Table 1 is shown in Figure 1.

Every leaf in the decision tree represents a classification rule. The path from the root of the tree to a leaf determines the conditions of the corresponding rule. The class at the leaf represents the rule's action.

Note that the critical decision in such a top-down decision tree generation algorithm is the choice of attribute at a node. The attribute selection is based on minimizing an information entropy measure applied to the examples at a node. The measure favors attributes that result in partitioning the data into subsets that have low class entropy. A subset of data has low class entropy when the majority of examples in it belong to a single class. The algorithm basically chooses the attribute that provides the locally maximum degree of discrimination between classes. For a detailed discussion of the information entropy minimization selection criterion see [Quin86, Fayy91].

3. OVERCOMING PROBLEMS WITH ID3 TREES

It is beyond the scope of this paper to discuss the details of the ID3 algorithm and the criterion used to select the next attribute to branch on. The criterion for choosing the attribute clearly determines whether a "good" or "bad" tree is generated by the algorithm[1]. Since making the optimal attribute choice is computationally infeasible, ID3 utilizes a heuristic criterion which favors the attribute that results in the

[1] See [Fayy90,Fayy91] for the details of what we formally mean by one decision tree being better than another.

partition having the least information entropy with respect to the classes. This is generally a good criterion and often results in relatively good choices. However, there are weaknesses inherent in the ID3 algorithm that are due mainly to the fact that it creates a branch for each value of the attribute chosen for branching.

3.1. PROBLEMS WITH ID3 TREES

Let an attribute A, with values { $a_1,a_2,...,a_r$ } be selected for branching. ID3 will partition the data along r branches each representing one of the values of A. However, it might be the case that only values a_1 and a_2 are of relevance to the classification task while the rest of the values may not have any special predictive value for the classes. These extra branches are harmful in three ways:

1. They result in rules that are overspecialized. The leaf nodes that are the descendants of the nodes created by the extraneous branches will be conditioned on particular irrelevant attribute values.
2. They unnecessarily partition the data, thus reducing the number of examples at each child node. The subsequent attribute choices made at such child nodes will be based on an unjustifiably reduced subset of data. The quality of such choices is thus unnecessarily reduced.
3. They increase the likelihood of occurrence of the missing branches problem. This problem occurs because not every possible combination of attribute values is present in the examples.

The third problem can be illustrated in the ID3 tree shown in Figure 1. Consider two possible unclassified examples which are to be classified by the tree of Figure 1:

ex1: (S = low) & (L = low)
ex2: (S = normal) & (L = low)

Both ex1 and ex2 are examples that have combinations of attribute values that did not appear in the training set of Table 1. However, the tree readily classifies ex1 as being of class *FRL*, but ex2 fails to be classified by the tree. This is because the subtree under the (S = normal) branch has no branch for (L = low). We shall shortly illustrate how the occurrence of *missing branches* may be avoided.

The main problem with the tree of Figure 1 is that the normal and high S branches should not be separated. Low S is the only value of relevance to the occurrence of a *LFR* event. It would be desirable if the learning algorithm could somehow take account of such situations by avoiding branching on attribute values that are not individually relevant. This would reduce the occurrence of the three problems listed above.

3.2. ALGORITHMS GID3* AND O-BTREE

As discussed earlier, improving the tree generation algorithm will improve the classification accuracy of the produced classifier. To avoid some of the problems described in the previous section, we developed the GID3* algorithm [Fayy91]. We generalized the ID3 algorithm so that it does not necessarily branch on each value of the chosen attribute. GID3* can branch on arbitrary individual values of an attribute and "lump" the rest of the values in a single *default branch*. Unlike the other branches of the tree which represent a single value, the default branch represents a subset of values of an attribute. Unnecessary subdivision of the data may thus be reduced. Figure 2 shows the tree GID3* would generate for the data set of Table 1. Note that both examples, ex1 and ex2, are classifiable by this tree. The missing branch problem that prevented the tree of Figure 1 from classifying ex2 does not occur in the tree of Figure 2.

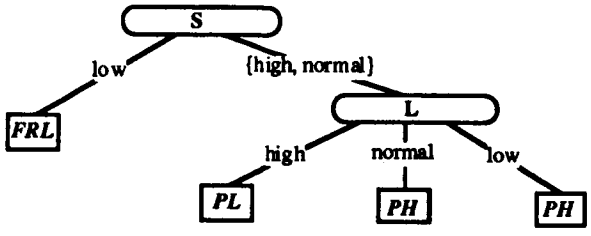

Figure 2: Decision Tree generated by GID3* for Data of Table 1.

The other learning algorithm we use is O-BTree [Fayy91,Fayy92b]. Unlike ID3 and GID3* whose attribute selection criterion is based on information entropy, O-BTree's selection criterion employs a measure from a different family of selection measures that measure class separation rather than class impurity (as in entropy). For a detailed discussion of GID3* and O-BTree as well as extensive empirical demonstration of their comparative performance, see [Fayy91].

Note that in our discussion and examples we assumed that attributes are discrete. Numerical attributes are discretized prior to attribute selection at each node. The range of a numerical attribute is partitioned into

several intervals thus creating a (temporary) discrete attribute. For a detailed discussion of attribute discretization see [Fayy91, Fayy92a]. Interested readers are referred to [Fayy91] for detailed accounts of the algorithms, the attribute selection criteria, the weaknesses of the ID3 approach, and various performance measures to evaluate the quality of the resulting trees. Performance measures include error rate in classifying new examples and measures of the size complexity of the generated tree.

We turn our attention to the task of automating sky object classification. We then present our results that demonstrate that O-BTree and GID3* performed significantly better than ID3 for this domain.

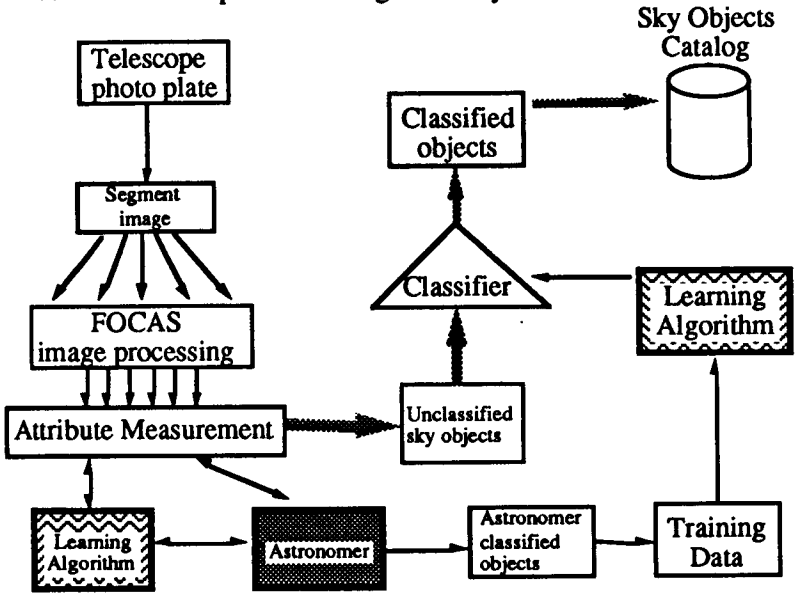

Figure 3. Architecture of the SkICAT System.

4. Classifying Sky Objects

Due to the large amounts of data being collected, a manual approach to classifying sky objects in the images is infeasible (it would require on the order of tens of man years). Existing computational methods for processing the images will preclude the identification of the majority of objects in each image since they are at levels too faint for traditional recognition algorithms or even manual inspection/analysis approaches. Our main objective is to provide an effective, objective, and examinable basis for classifying sky objects.

The photographic plates collected from the survey are being digitized at the Space Telescope Science Institute (STScI). This process will result in about 1788 digital images of roughly $23{,}000^2$ pixels each. Low-level image processing and object separation is performed by the FOCAS image processing software developed at Bell Labs [Jarv81,Vald82]. In addition to defining the objects in each image, FOCAS also produces basic attributes describing each object. A digitized plate is subdivided into a set of partially overlapping frames. Each frame represents a small part of the plate that is small enough to be manipulated and processed conveniently. Figure 3 depicts the overall architecture of the proposed SkICAT System. The discussion below will explain the loop in the bottom left-hand corner in which machine learning is employed in the attribute measurement process. The image processing steps that a digitized plate goes through are:

1. Select a frame from the digitized plate.
2. Detection: detect contiguous pixels in the image that are to be grouped as one object (standard image processing).
3. Perform more accurate local sky determination for each detected object.
4. Evaluate parameters for each object independently: we initially measured 18 *base-level attributes*.
5. Split objects that are "blended" together and re-evaluate attributes.
6. **AUTOPSF**: select a subset of the objects in the frame and designate them as being "sure-thing stars, form PSF template.
7. Measure *resolution scale* and *resolution fraction* attributes for each object: These are obtained by fitting the object to the template of sure-thing stars formed in step 6.
8. Classify objects in image.

All steps are automated except for steps 6 and 8. Step 6 needs further elaboration. The goal of this step is to define the two resolution attributes mentioned in step 7. These attributes are parameters of a template defined on a point spread function (PSF). The template is computed over a subset of objects identified as sure-thing stars. The sure-thing stars are selected by the astronomer. They represent the "archetypal" stars in that image. Once the stars are selected, the template fitting and resolution parameter measurements are computed automatically. Thus in order to automate steps 1-7 we need to automate the star selection step (6). We refer to this problem as the *star selection subproblem.*

The 18 base-level attributes measured in step 4 are [Vald82]:

- isophotal magnitude
- isophotal area
- core magnitude
- core luminosity
- sky brightness
- orientation
- sky sigma (variance)
- image moments (8): ir1, ir2, ir4, r1, r2, ixx, iyy, ixy.
- eccentricity (ellipticity)
- semi-major axis
- semi-minor axis.

Once all attributes, including the resolution attributes, for each object are measured, step 8 involves performing the final classification for the purposes of the catalog. We are currently considering classifying objects into four major categories: star (s), star with fuzz (sf), galaxy (g), and artifact (long). We may later refine the classification into more classes, however, classification into one of the four classes represents our initial goal.

4.1. CLASSIFYING FAINT OBJECTS AND THE USE OF CCD IMAGES

In addition to the scanned photographic plate, we have access to CCD images that span several small regions in some of the frames. CCD images are obtained from a separate telescope. The main advantage of a CCD image is higher resolution and signal-to-noise ratio at fainter levels. Hence, many of the objects that are too faint to be classified by inspection of a photographic plate, are easily classifiable in a CCD images. In addition to using these images for photometric calibration of the photographic plates, we make use of CCD images in two very important ways for the machine learning aspect:

1. CCD images enable us to obtain class labels for faint objects in the photographic plates.
2. CCD images provide us with the means to reliably evaluate the accuracy of the classifiers obtained from the decision tree learning algorithms.

Recall that the image processing package FOCAS provides the measurements for the base-level attributes (and the resolution attributes after star selection) for each object in the image. In order to produce a classifier that classifies faint objects correctly, the learning algorithm needs training data consisting of faint objects labeled with the appropriate class. The class label is therefore obtained by examining the CCD frames. Once trained on properly labeled objects, the learning algorithm produces a classifier that is capable of properly classifying objects based on the values of the attributes provided by FOCAS. Hence, in principle, the classifier will be able to classify objects in the photographic image that are simply too faint for an astronomer to classify by inspection. Using the class labels, the learning algorithms are basically being used to solve the more difficult problem of separating the classes in the multi-dimensional space defined by the set of attributes derived via image processing. This method is expected to allow us to classify objects that are at least one magnitude fainter than objects classified in photographic sky surveys to date.

4.2. INITIAL RESULTS FOR THE CLASSIFICATION PROBLEM

Starting with digitized frames obtained from a single digitized plate, we performed initial tests to evaluate the accuracy of the classifiers produced by the machine learning algorithms ID3, GID3*, and O-BTree. The data consisted of two frames from a single plate. The two frames were chosen such that we had a CCD counterpart for each of them. The first frame contains the Abell 68 cluster of galaxies (A68) and the second frame contains the Abell 73 cluster (A73). A68 has 88 objects and A73 has 96. We trained the algorithms on training data from one frame and then used the second frame to independently evaluate the accuracy of the decision tree produced. The results may be summarized as follows: Algorithms GID3* and O-BTree produced significantly better trees than ID3. Accuracy for GID3* was about 90%. O-BTree's accuracy was slightly better and the trees generated by O-BTree were on average more compact (smaller number of leaves). O-BTree produced trees that on average had about 6 leaves.

In contrast, the best ID3 tree had 10 leaves and an error rate of 20.5%.

However, when the same experiments were conducted without using the *resolution scale* and *resolution fraction* attributes of step 6, the results were significantly worse. The error rates jumped above 20% for O-BTree, above 25% for GID3*, and above 30% for ID3. The respective sizes of the trees grew as well.

The initial results may be summarized as follows:

1. Algorithms GID3* and O-BTree produced significantly better trees than ID3.
2. Classification accuracy results of better than 90% were obtained when using two user-defined attributes: *resolution fraction* and *resolution scale.*
3. Classification results were not as reliable and stable if we exclude the two resolution attributes.

We took this as evidence that the resolution attributes are very important for the classification task. Hence we turned to addressing the star selection subproblem in order to automate step 6 above. Furthermore, the results point out that the GID3* and O-BTree learning algorithms are more appropriate than ID3 for the final classification task.

4.3. AUTOMATING THE STAR SELECTION PROCESS

Based on the initial results of the previous section, it was determined that using the resolution attributes is necessary since without them the error rates were significantly worse. We do not have the option of leaving star selection as a manual step in the process, since it is a time consuming task and will easily become the bottleneck in the system. We decided to use a machine learning approach to solve the star selection subproblem.

The star selection subproblem is a binary classification problem. Given a set of objects in an image, the goal is to classify them as sure-thing stars and non-sure-thing stars. Using the data from A68 and A73 we were able to obtain better than 94% accuracy on selecting stars. However, these data sets came from a single plate. We also needed to evaluate the robustness of the produced classifiers when going across plates: i.e. test the classifier on images from plates other than the plate from which the training data was drawn. Since we had access to only one plate scanned by STScI, we decided to use plates scanned by a lower resolution scanner: the COSMOS scanner at the Royal Observatory, Edinburgh (ROE). We obtained frames from three different POSS-II plates: F2268, F2249, and F830.

Although our cataloguing task will eventually use STScI scans, we decided to use the COSMOS scans to conduct our initial testing. It is strongly believed that the results obtained on the COSMOS scan will be lower bounds on the performance attainable on the higher quality STScI scans. We constructed training data using subsamples of F2268 and F2249. Data from F830 were to be used strictly for testing purposes.

The data objects from all three plates were classified manually by one of the authors (N. Weir) into *sure-stars*, *non-sure-stars*, and *unknowns*. The goal of the learning subproblem is to construct classifiers for selecting out sure-stars from any collection of sky objects.

Although our accuracy on classifying data from the same plate was around 94%, the accuracy dropped to 60%-80% levels when classifying data from different plates. It was determined that the base-level attributes such as area, background-sky-levels, and average intensity are image-dependent as well as object-dependent. It was also determined that a new set of user-defined attributes needed to be formulated. These attributes were to be computed automatically from the data, and are defined such that their values would be normalized across images and plates. It is beyond the scope of this paper to give the detailed definitions of these new attributes.

4.4. CROSS-PLATE ROBUSTNESS & COMPARISON WITH NEURAL NETS

As expected, defining the new "normalized" attributes raised our performance on both intra- and inter-plate classification to acceptable levels varying between 92% and 98% accuracy. We expect the results to be better for higher resolution STScI scans, but we have not yet verified this.

In order to compare against other learning algorithms, and to preclude the possibility that a decision tree based approach is imposing *a priori* limitations on the achievable classification levels, we tested several neural network algorithms for comparison. The results indicate that neural network algorithms achieve

similar, and sometimes worse performance than the decision trees. The neural net learning algorithms tested were: traditional backpropagation, conjugate gradient optimization, and variable metric optimization. The latter two are training algorithms that work in batch mode and use standard numerical optimization techniques in changing the network weights. Their main advantage over backpropagation is the significant speed-up in training time.

Upon examining the results of the empirical evaluation, we concluded that the neural net approach did not offer any clear advantages over the decision tree based learning algorithm. Although neural networks, with extensive training and several training restarts with different initial weights to avoid local minima, could match the performance of the decision tree classifier, the decision tree approach still holds several major advantages. The most important is that the tree is easy for domain experts to understand. In addition, unlike neural network learning algorithms, the decision tree learning algorithms GID3* and O-BTree do not require the specification of parameters such as the size of the neural net, the number of hidden layers, and random trials with different initial weight settings. Also, the required training time is orders of magnitude faster than the training time required for a neural network approach.

4.5. VERIFICATION AND RELIABILITY ESTIMATES

As mentioned earlier, in addition to using the CCD frames to derive training data for the machine learning algorithms, we also use them to verify and estimate the performance of our classification technique. This is done by testing on data sets that are drawn independently from the training data. An additional source of internal consistency checks comes from the fact that the plates, and the frames within each plate are partially overlapping. Hence, objects inside the overlapping regions will be classified in more than one context. By measuring the rate of conflicting classifications, we can obtain further estimates of the statistical confidence in the accuracy of our classifier. For the purposes of the final catalog production, a method needs to be designed for resolving conflicts on objects within regions of overlap. We have not yet addressed this question.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we gave a brief overview of the use of machine learning techniques for automatically producing classification decision trees. We motivated the problem addressed by machine classification learning in a general context as well as in the particular context of our application domain: the automation of sky object catalog generation. If successful, the SkICAT system is expected to speed up catalog generation by one to two orders of magnitude over traditional manual approaches to cataloguing. This should significantly reduce the cost of cataloguing survey images by the equivalent of tens of astronomer man-years. In addition, we aim to classify objects that are at least one magnitude fainter than objects catalogued in previous surveys. Finally, this project represents a step towards the development of an objective, reliable automated sky object classification method.

The initial results of our effort to automate sky object classification in order to automatically reduce the images produced by POSS-II to sky catalogs are very encouraging. With the use of the resolution attributes we expect to have an accuracy at or above 90%. Since measurement of the resolution attributes requires interaction with the user in selecting sure-thing stars for template fitting, we used the same machine learning approach to automate the star selection process. By defining additional "normalized" image-independent attributes, we were able to obtain high accuracy classifiers for star selection within and across photographic plates. This in turn allows us to automate the computation of the powerful resolution attributes for each object in an image. This is taken as a strong indication that our automated cataloguing scheme will ultimately achieve the desired accuracy levels.

The positive initial results obtained for the star selection subproblem suggest pursuing the derivation of more image-independent attributes to describe the sky objects in an image. This is expected to lead to higher levels of classification accuracy as well as more robust classifiers. In addition, we plan to extend the basic decision tree approach we are using to one that is based on learning statistically robust rules. This approach would be based on generating multiple decision trees and selecting the best rules out of each tree. The rules are eventually merged to obtain optimal coverage of the training data sets. In our experience with the decision tree algorithms, we noticed that the decision tree produced by the learning algorithm typically has leaves which represent overspecialized or incorrect classification rules. This suggests that an overall better classifier can be constructed by generating multiple trees and selecting only good rules from each. We have adopted this methodology in the past in other domains and it has been our experience that more compact and more robust classifiers are obtained [Chen90]. We expect that adopting it here will further improve performance.

Final object classification will be, to some extent, also a matter of scientific choice. While objects in every catalog will contain a classification entry, all of the object attributes will be recorded as well. One could therefore reclassify any portion of the survey using alternative criteria better suited to a particular scientific goal (e.g. star catalogs vs. galaxy catalogs). The catalogs will also accommodate additional attribute entries, in the event other pixel-based measurements are deemed necessary. An important feature of the survey analysis system will be to facilitate such detailed interactions with the catalogs.

As part of our plans for the future we also plan to begin investigation of the applicability of unsupervised learning (clustering) techniques such as AUTOCLASS [Chee88] to the problem of discovering clusters or groupings of interesting objects. The goal is to evaluate such a capability as an aid for the types of analyses astronomers conduct after objects have been classified into known classes. Typically, astronomers examine the various distributions of different types of objects to test existing models of the formation of large-scale structure in the universe. Armed with prior knowledge about properties of interesting clusters of sky objects, a clustering system can search through catalog entries and point out potentially interesting object clusters to astronomers. This will help astronomers catch important patterns in the data that may otherwise go unnoticed due to the sheer size of the data volumes.

ACKNOWLEDGMENTS

We would like to thank the COSMOS group at the ROE, in particular H. T. MacGillivray for providing the scan data for this work. Also, many thanks to the Sky Survey team for their expertise and effort in acquiring the plate material. The POSS-II is funded by grants from the Eastman Kodak Company, The National Geographic Society, The Samuel Oschin Foundation, NSF Grants AST 84-08225 and AST 87-19465, and NASA Grants NGL 05002140 and NAGW 1710. We thank Joe Roden for help on evaluating the performance of the learning algorithms. This work was supported in part by a NSF graduate fellowship (N. Weir), the Caltech President's Fund, NASA contract NAS5-31348 (S. Djorgovski and N. Weir), and the NSF PYI Award AST-9157412 (S. Djorgovski).

The work described in this paper was carried out in part by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[Brei84] Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and Regression Trees.* Monterey, CA: Wadsworth & Brooks.

[Chee88] Cheeseman, P., Self, M., Kelly, J., Taylor, W. Freeman, D. and Stutz, J. (1988) "Bayesian Classification." *Proceedings of the Seventh National Conference on Artificial Intelligence AAAI-88*, pp. 607-6611, Saint Paul, MN.

[Chen90] Cheng, J., Fayyad, U.M., Irani, K.B. and Qian, Z. (1990) "Applications of machine learning techniques in semiconductor manufacturing." *Proceedings of the SPIE Conference on Applications of Artificial Intelligence VIII.* pp. 956-965, Orlando, Fl.

[Feig81] Feigenbaum, E.A. (1981) "Expert systems in the 1980s." In Bond, A. (Ed.) *State of The Art Report on Machine Intelligence.* Maidenhead: Pergamon-Infotech.

[Fayy90] Fayyad, U.M. and Irani, K.B. (1990). "What should be minimized in a decision tree?" *Proceedings of Eighth National Conference on Artificial Intelligence AAAI-90*, Boston, MA.

[Fayy91] Fayyad, U.M. (1991). *On the Induction of Decision Trees for Multiple Concpt Learning.* PhD Dissertation, EECS Dept. The University of Michigan.

[Fayy92a] Fayyad, U.M. and Irani, K.B. (1992) "On the handling of continuous-valued attributes in decision tree generation." *Machine Learning*, vol.8, no.2.

[Fayy92b] Fayyad, U.M. and Irani, K.B. (1992) "The attribute selection problem in decision tree generation." *Proceeedings of the Tenth National Conference on Artificial Intelligence, AAAI-92.* San Jose, CA.

[Jarv81] Jarvis, J. and Tyson, A. (1981) *Astronomical Journal* 86:41.

[Quin86] Quinlan, J.R. (1986) "The induction of decision trees." *Machine Learning* vol. 1, no. 1.

[Vald82] Valdes (1982) *Instrumentation in Astronomy IV*, SPIE vol. 331, no. 465.