The Decision Tree Classifier: Design and Potential

PHILIP H. SWAIN, MEMBER, IEEE, AND HANS HAUSKA

Abstract—**This paper presents the basic concepts of a multistage classification strategy called the decision tree classifier. Two methods for designing decision trees are discussed and experimental results are reported. The relative advantages and disadvantages of each design method are considered. A spectrum of typical applications in remote sensing is noted.**

I. Introduction

A RESULT of the launch of two Landsat satellites in this decade has been an enormous increase in the volume of available multispectral remote sensing data and a growing interest in machine analysis of such data. Many of the potential applications of the data require more efficient numerical analysis techniques than those which have most commonly been utilized. In particular, conventional maximum likelihood (ML) classifiers are characterized by two significant drawbacks.

1) Only one of the possible combinations of pattern features is used in the classification.

2) Each data sample is tested against all classes in a classification, which leads to a relatively high degree of inefficiency.

Another problem often encountered is the so-called dimensionality problem. With a fixed relatively small size training set the classification accuracy may actually decrease when the number of features is increased [1]. Such a constraint on the training data is very common when working with Landsat data.

Another inherent weakness of the ML procedure is that the subset of pattern features used in a classification is not necessarily the optimum for all the classes. Usually the set of pattern features is selected by the criterion of maximum *average* interclass separability, i.e., in a multiclass multifeature classification the set of pattern features for which the *average* pairwise separability is largest will be used.

The number of tests necessary in a multiclass multifeature classification can often be significantly reduced using a sequence of tests. Several types of multistage classification schemes are known [2]. It is the purpose of the present paper to discuss a class of multistage classifiers which we call decision tree classifiers.

The decision tree classifier is characterized by the fact that an unknown sample is classified into a class using one or several decision functions in a successive manner. This classification strategy can be described by a tree diagram (Fig. 1). For processing purposes the tree is encoded as a string of symbols such that there is a unique relationship between the string and decision tree. The string is decoded in the computer and pointers are set up to define the appropriate classification path for each data sample.

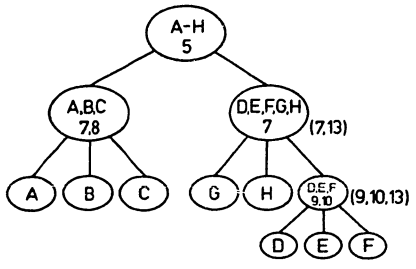

Fig. 1. A simple decision tree, illustrating terminology.

In general, a decision tree consists of a root node, a number of interior nodes, and a number of terminal nodes. The root node and interior nodes, referred to collectively as nonterminal nodes, are linked into decision stages; the terminal nodes represent final classifications. (See Fig. 1.) Associated with the root node is the entire set of classes into which a sample may be classified. The set of nodes at a given level in the tree, i.e., all the same "distance" from the root, is called a layer. Each node consists of a set of classes to be discriminated, the set of features to be used, and the decision rule for performing the classification.

In this paper we shall discuss two methods for designing decision trees, present some experimental results, and consider the outlook on the potentials of this classification approach for a variety of multispectral and multitemporal remote sensing applications.

II. The Design of Decision Trees

To achieve the best possible performance with a classifier as described above, the design of the decision stages is of utmost importance. The choice of tree structure and the choice of appropriate feature subsets will be reflected in the performance (classification accuracy) and efficiency (computer time used for the classification). For the purposes of this paper, we shall restrict the decision rule at each stage to be the Gaussian maximum likelihood rule. In the following sections we will discuss two approaches for the design of decision trees. These approaches are similar in principle, but differ significantly in the way the tree is actually designed.

A. The Manual Design Procedure

After the statistics for all the classes have been computed (means and covariance matrices), a graph is obtained in which the means and variances for all the classes are plotted for each feature. It is then possible to estimate from this graph suitable decision boundaries such that all classes are separated in a number of decision steps. As long as we restrict the number of

Manuscript received March 1, 1975; revised February 28, 1977. This work was supported in part by the National Aeronautics and Space Administration under Contracts NAS9-14016 and NAS9-14970.

P. H. Swain is with the School of Electrical Engineering and Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, IN 47907.

H. Hauska was with the Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, IN 47907. He is now with Lulea University of Technology, S-95187 Lulea, Sweden.

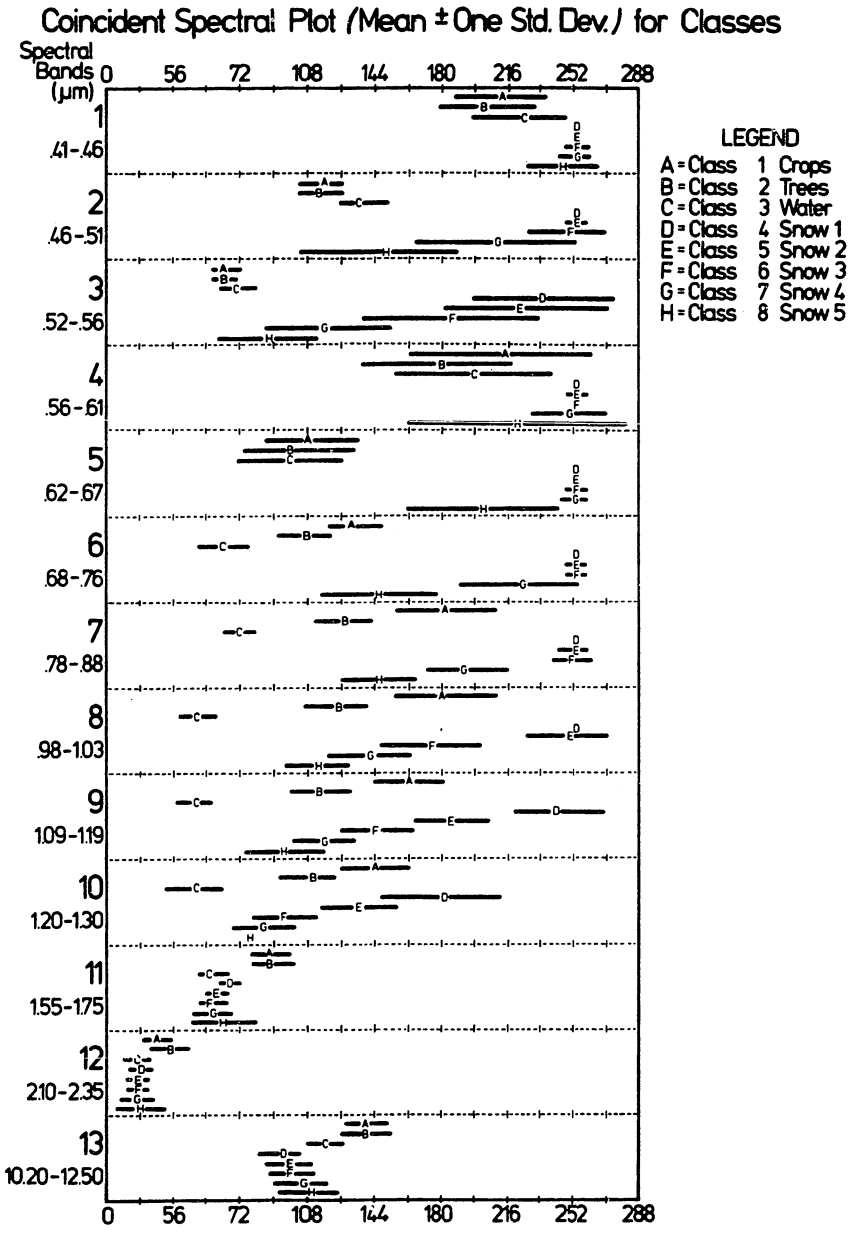

Fig. 2. Coincident spectral plot for the San Juan Mountains example.

features used in each stage to one, this is roughly equivalent to estimating a simple distance measure between classes. The method is not suitable, however, when two or more features are to be used in a stage of the tree, because the graph does not show how the interactions between features can be used to advantage.

To illustrate the procedure, Fig. 2 shows a "coincident spectral plot" for an eight-class 13-feature classification. The data were taken by Skylab, using a multispectral scanner, on June 5, 1973, over a test site in the San Juan Mountains, CO. We have chosen this particular example because it demonstrates both advantages and disadvantages of this approach. A first inspection of the figure shows that the statistics for classes *D*, *E*, and *F* are ill-conditioned; that is, in a one-stage maximum likelihood classification scheme these feature sets could not be used because a zero variance indicates a singular covariance matrix for these classes. Also, the computation of any separability measure based on second-order statistics (such as divergence or Bhattacharyya distance) would be inhibited due to problems in matrix inversion. In the multistage approach to classification, these features can be used to discriminate among classes as long as they are not used to classify those specific classes for which they would result in a singular covariance matrix, and, in fact, this was desirable in the example we are discussing.

As seen in Fig. 2, feature 5 is well suited for separating class subset (*A*, *B*, *C*) from subset (*D*, *E*, *F*, *G*, *H*). The first layer of the decision tree is thereby determined. We must then look for a feature set which can classify the mixture of classes (*A*, *B*, *C*) into single classes. Feature 7 or 8 could be selected. The mixture (*D*, *E*, *F*, *G*, *H*) must also be divided into its components. Feature 7 will discriminate among the mixture (*D*, *E*,

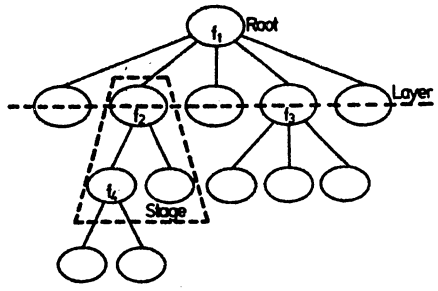

Fig. 3. Manually designed decision tree for the San Juan Mountains example (parentheses indicate final feature selection).

F) and classes *G* and *H*, although use of this feature requires pooling of classes *D*, *E*, and *F* to avoid the singular covariance matrix. Finally, to separate classes *D*, *E*, and *F*, feature 9 or 10 could be used, but, as the result of a test classification, it was decided to use both to maximize accuracy. Similarly, it was decided to use both features 7 and 8 to discriminate among classes *A*, *B*, and *C*. The decision tree is shown in Fig. 3.

The results obtained by classifying the data based on this tree were good, but some of the errors observed seemed to be related to a correlation of classes *D*, *E*, *F*, *G*, and *H* and the topography. To improve the discriminability of these classes, another feature was added, modifying the decision tree as indicated in Fig. 3. The classification improved sufficiently to merit the added computational cost, especially due to the analyst's particular interest in classes *D*, *E*, *F*, *G*, and *H*.

Notice that to get roughly the same results for this classification problem using the conventional single-stage classifier would require use of six-variable statistics to classify every data point. Since the computational cost is roughly proportional to the square of the number of features used, the decision tree classifier was much more efficient for this problem—even with the overhead required to realize the multistage logic.

The coincident spectral plot (Fig. 2) provides an estimate of the interclass separability based on single features. If the difficulty of discriminating the classes requires use of a combination of several features, the manual design approach based on the spectral plot is severely limited. A highly skilled analyst familiar with the multivariate interactions in the data can, for some cases, use a trial-and-error approach and derive a suitable decision tree. But, in general, a more analytical multivariate design procedure is desirable when the complexity of the problem—in terms of the number of classes involved or the number of features required for adequate classification accuracy—is significant. We will discuss such a procedure next.

B. Toward Optimized Decision Tree Design

Ideally, we would like to have a procedure to determine for any given problem the optimal decision tree, i.e., the tree which defines the classifier achieving the highest possible classification accuracy while requiring the smallest possible computation time. As we have already suggested, the manual design procedure provides optimal results only in extremely simple cases (relatively few classes, easily discriminated with a small number of features). For more complex problems, we design the decision tree using a heuristic search technique described as "guided search with forward pruning" [4]. This method uses an evaluation function to direct a search through the possible decision tree structures such that, at each stage, the node configuration selected is the one with the highest evaluation function.

The evaluation function used in the search is a weighted measure of classifier efficiency and accuracy; i.e., the evaluation function for a given node d_i is of the form

$$E(d_i) = -T(d_i) - K \cdot \epsilon(d_i) + \sum_{j=1}^{c_i} P_{i+j} E(d_{i+j}) \tag{1}$$

where $T(d_i)$ is the computation time and $\epsilon(d_i)$ is the classification error associated with the node; $E(d_{i+j})$ is the evaluation function associated with the jth descendant node of d_i; node d_i has c_i descendant nodes; and P_{i+j} is the probability that the jth descendant node will be reached from d_i. The constant K is specified by the user to express the relative importance of classifier speed versus accuracy. One candidate structure for node d_i is the conventional classifier. Certainly we are not interested in any candidate structure which does not perform as well as a conventional classifier. For the conventional structure, we write the evaluation function as

$$E_0(d_i) = -T_0(d_i) - K \cdot \epsilon_0(d_i) \tag{2}$$

(note that all descendant nodes for this structure are terminal nodes). We then write a "normalized" node evaluation function as

$$\begin{aligned} E'(d_i) &= E(d_i) - E_0(d_i) \\ &= [T_0(d_i) - T(d_i)] + K[\epsilon_0(d_i) - \epsilon(d_i)] \\ &\quad + \sum_{j=1}^{c_i} P_{i+j} E(d_{i+j}). \end{aligned} \tag{3}$$

Now a candidate node configuration remains a candidate only if $E'(d_i)$ is positive. Finally, since we are using a forward search procedure, we do not yet know the configurations of the descendants of node d_i. We shall therefore approximate their evaluations by the lower bound $E_0(d_{i+j})$. The final form of the evaluation function for a node is thus

$$\begin{aligned} E''(d_i) &= [T_0(d_i) - T(d_i)] + K[\epsilon_0(d_i) - \epsilon(d_i)] \\ &\quad + \sum_{j=1}^{c_i} P_{i+j} E_0(d_{i+j}). \end{aligned} \tag{4}$$

Decision trees designed by applying E'' node-by-node are almost certainly suboptimal in the sense that there is no assurance that for the resulting tree $E(d_1)$ will be maximized. Truly optimal decision tree design is an exceedingly complex problem and still the subject of research. We have, however, developed a heuristic search procedure based on E'' and demonstrated experimentally that it provides a useful tool for deriving accurate and efficient decision tree classifiers. For recent research results pertaining to decision tree optimization, see [5].

A series of experiments were performed in which a number of decision tree classifiers were designed using this heuristic search procedure. The data were taken by the multispectral

TABLE I
COMPARISON OF DECISION TREE CLASSIFIERS AND SINGLE-STAGE CLASSIFIERS

MAXIMUM NUMBER OF FEATURES PER NODE	SEPARABILITY MEASURE	SEPARABILITY THRESHOLD	TRADE OFF CONSTANT	DECISION TREE CLASSIFICATION TIME (SECONDS)	DECISION TREE TRAINING FIELD ACCURACY (%)	SINGLE STAGE CLASSIFICATION TIME (SECONDS)	SINGLE STAGE TRAINING ACCURACY (%)
4	B_T	1950	20.0,10.0	545	93.5	1574	93.7
4	D_T	1950	10.0	655	93.6		
3	B_T	1950	10.0	440	92.9	1036	93.0
3	D_T	1950	25.0	520	92.9		
2	B_T	1950	5.0	450	91.6	650	90.2
2	B_T	1850	5.0	390	90.4		
2	D_T	1850	5.0	435	92.2		

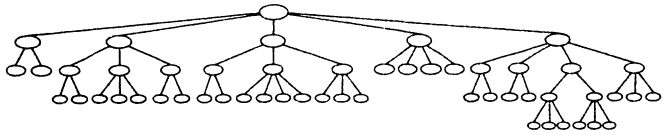

Fig. 4. A typical machine-designed decision tree for the Kenosha Pass example (corresponds to top row of Table I).

scanner aboard Landsat-1 over Kenosha Pass, CO, on August 15, 1973 (frame no. 1388-17134). The classification problem involved 19 classes and four features.

A Gaussian ML decision rule was used at each node of the decision tree. Because of the problems inherent in directly estimating the error probability associated with such a decision rule, interclass separability was used instead. Both transformed divergence (D_T) and a transformed Bhattacharyya distance (B_T) were used as measures of interclass separability [4], [6]. Results to date do not indicate a significant advantage for either of these separability measures.

The performances of the decision tree classifiers were compared to the performances of single-stage ML classifiers implemented with the same software. The single-stage classifiers used the union of all features appearing in the decision tree. As seen in Table I, all of the resulting decision tree classifiers were considerably more efficient while yielding virtually identical accuracy (for this data/problem situation, accuracy could not be significantly improved by using multistage logic). A typical tree structure designed by the heuristic search procedure is shown in Fig. 4.

The heuristic tree design procedure is a useful tool for designing decision tree classifiers in situations too complex to be dealt with effectively by manual means such as described in the preceding section. But since the procedure does not produce the optimal tree, it is a good idea to develop a number of decision tree classifiers by systematically varying the parameters at the user's disposal (the tradeoff constant, the separability criterion, the separability threshold). The performance of each classifier can then be determined by classifying the set of training data.

C. Some Procedural Details

The following general procedure has evolved through our experience with decision tree classifier design. It assumes that adequate reference data, or "ground truth," are available to characterize the classes of interest. It also assumes that the decision rule used at each node is a Gaussian maximum likelihood rule, although generalization would not be difficult.

1) From training samples, compute the mean vectors and covariance matrices for the classes of interest. Alternatively, these may be derived by more complex classifier training procedures such as those described in [7]. Use a measure of classifier separability to decide whether it is really feasible to discriminate all of the classes or whether some should be merged or deleted from further consideration.

2) Use a feature selection algorithm to determine a subset of the available features to be considered for use in the decision tree classifier. Minimizing the size of this subset will improve the efficiency of the design procedure, whether the manual procedure or the heuristic search procedure is used. However, no feature should be deleted which is necessary to adequately discriminate any class of interest. At this point, the feature selection criterion should be based on pairwise separability over all pairs of classes. Save the class separability information for use in the tree design process.

3) Use the manual design procedure or the heuristic search procedure to design the decision tree or a set of candidate trees.

4) Draw the decision tree(s) and code appropriately for classification.

In general, the breadth of the tree will reflect the relative weight given to classifier accuracy whereas the depth of the tree will reflect the weight given to efficiency. Broad trees are typically characterized by the use of several features at each node, which tends to increase accuracy; deep trees use a very small number of features at each node, which reduces computational complexity at the node.

Having drawn the tree(s) for visual inspection, the experienced remote sensing data analyst may make use of insights he may have concerning energy-matter interactions and the specific problem at hand to make final adjustments to the decision tree, typically near the terminal nodes. Although these adjustments "tidy up" the tree and make the classifier somewhat simpler to implement, they will often not impact significantly the classifier accuracy or efficiency.

III. Applications

In our introductory discussion we noted that decision tree classifiers offer efficiency and accuracy advantages over conventional single-stage classifiers. More than this, however, the flexibility of this classifier model enhances its applicability to a wide range of problems to which the conventional classifiers are at best awkwardly applied. A number of such problems are listed in Table II, and we shall discuss them briefly.

Multi-Image Analysis: There is a wide range of remote sensing applications for which the use of multiple images of the scene is necessary or desirable (i.e., images other than those collected simultaneously by a given sensor on a single pass). For example, through multitemporal analysis, it is possible to utilize the information contained in differential rates of change of various ground cover types to characterize those cover types. Naively, one might think it would be sufficient, once the images have been precisely registered, to simply concatenate the data vectors from the various images (e.g., make an eight-feature vector from the two 4-channel pixels recorded on two successive Landsat passes) and apply the same analysis techniques normally applied to data from a single image. This approach is only occasionally successful, however, largely for the following two reasons. First, if for any reason the data for a given pixel are unavailable in one or more of the images (due to clouds or data system errors, for instance), the pixel becomes unclassifiable. Second, the number of spectral/temporal subclasses into which any single class must be subdivided (and hence the number of subclasses which must be characterized by training data) tends to be proportional to the product of the number of subclasses in the individual images. In practice one finds that the number of subclasses which must be accounted for quickly outpaces the amount of reference data available for training the classifier.

The decision tree classifier simplifies the situation substantially because the analysis of the multiple images can be "decoupled"; that is, at any node of the tree, the features used can be limited to those from a single image. As a result, the number of subclasses increases as the sum of the number from the individual images. Furthermore, the dimensionality of the statistics used at any node will be minimized, further reducing the demand for reference data. Also, should the data for a given pixel be unavailable from some of the images, the decision tree logic can be formulated so as to make a classification anyway, based on what data is available.

TABLE II
Applications of Layered Classifiers

GENERAL APPLICATION	EXAMPLE
MULTI-IMAGE ANALYSIS	MULTITEMPORAL CLASSIFICATION CHANGE DETECTION
USE OF MIXED FEATURE TYPES	TEXTURE TOPOGRAPHY GEOPHYSICAL DATA
RECOGNITION OF CLASS-SPECIFIC PROPERTIES	CROP STRESS DETECTION FOREST TYPE MAPPING WATER QUALITY MAPPING WATER TEMPERATURE MAPPING

Change detection is often performed by classifying two registered images separately and then making a point-by-point comparison of the results. A decision tree classifier can be used to reduce this to a one-pass process. Still further efficiencies are gained by judicious tree design. For instance, if the user is interested only in changes from rural catagories to urban catagories, then a pixel classified other than rural in the "predecessor" data set can be ignored as of no further interest, without attempting classification of the "successor" data.

Use of Mixed Feature Types: With the development of data banks containing registered data from a variety of sources, it has now become possible to incorporate these data into the classification process. However, the statistical assumptions applied to purely multispectral data are rarely extendable to the multisource data. In fact, such extension may not be justified even when new features are derived from the multispectral data (texture features, for instance). The decision tree approach provides a means of constructing classifier logic consistent with the types of data available. Various nodes in the tree may utilize quite different types of classifiers compatible with the types of features associated with the node.

Class-Specific Properties: It is not surprising that the number of spectral features required to identify corn is, in general, smaller than the number required to discern that a given pixel is corn and also determine whether the corn is healthy or diseased. Similarly, a single feature may be adequate to identify water whereas several more may be needed to analyze the quality (turbidity, salinity, etc.) of the water. Certainly, then, to maximize classifier efficiency, we would want to use the additional features only in those cases where the more detailed discrimination was necessary, i.e., once the corn (or water) had already been identified. Clearly the decision tree classifier provides a mechanism for doing just that.

IV. Conclusions

The flexibility of the decision tree classifier makes it attractive for a wide range of applications, either for improving the classifier performance in general (maximizing accuracy while minimizing computational requirements) or for treating special applications in which multilevel decision logic is the only practical approach. Effective use of this approach requires, however, that means be available to determine a suitable decision tree for the problems at hand. In some cases, manual or interactive methods are adequate, although ideally one would like to have a computer-implemented algorithm capable of optimal tree design. Some success with the latter has been described here, but much remains to be done.

Acknowledgment

The authors wish to thank Dr. L. A. Bartolucci and M. Fleming for their contributions of the Skylab and Landsat analysis results based on the decision tree classifier. They are also indebted to the European Space Research Organization for their postdoctoral support of Dr. Hauska; and to Dr. C. L. Wu on whose original work their continuing research is based.

References

[1] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 55-63, Jan. 1968.
[2] K. S. Fu, *Sequential Methods in Pattern Recognition and Machine Learning*. New York: Academic Press, 1968.
[3] J. R. Slagle and R. C. T. Lee, "Application of game tree searching techniques to sequential pattern recognition," *Comm. ACM*, vol. 14, no. 2, Feb. 1971.
[4] C. Wu, D. Landgrebe, and P. Swain, "The decision tree approach to classification," School of Electrical Engineering, Tech. Rep. TR-EE 75-17, Purdue Univ., West Lafayette, IN, May 1975.
[5] A. V. Kulkarni and L. N. Kanal, "An optimization approach to hierarchical classifier design," in *Proc. Third Int. Joint Conf. Pattern Recognition* (Coronado, CA, Nov. 1976), IEEE Cat. No. 76CH1140-3C.
[6] P. H. Swain and R. C. King, "Two effective feature selection criteria for multispectral remote sensing," in *Proc. First Int. Joint Conf. Pattern Recognition* (Washington, DC, Nov. 1973).
[7] M. D. Fleming, J. S. Berkebile, and R. M. Hoffer, "Computer-aided analysis of LANDSAT-1 MSS data: A comparison of three approaches, including a 'modified clustering' approach," in *Proc. Third Symp. Machine Processing of Remotely Sensed Data* (Purdue Univ., June 1975) IEEE Cat. No. 75CH1009-0-C.

Estimation of the Probability of Error Without Ground Truth and Known *A Priori* Probabilities[1]

K. A. HAVENS, T. C. MINTER, AND S. G. THADANI

Abstract—The probability of error or, alternatively, the probability of correct classification (PCC) is an important criterion in analyzing the performance of a classifier. Labeled samples (those with ground truth) are usually employed to evaluate the performance of a classifier. Occasionally, the numbers of labeled samples are inadequate, or no labeled samples are available to evaluate a classifier's performance; for example, when crop signatures from one area from which ground truth is available are used to classify another area from which no ground truth is available. This paper reports the results of an experiment to estimate the probability of error using unlabeled test samples (i.e., without the aid of ground truth).

Manuscript received April 9, 1976. This work was supported by NASA under Contract NAS 9-12200 and was prepared for the Earth Observations Division, NASA/JSC, Houston, TX.

The authors are with Lockheed Electronics Co., Inc., Aerospace Systems Division, Houston, TX.

[1] In this paper, PCW and PCO will denote the probability of correctly classifying wheat and "other," respectively, as calculated by either the analytical procedure or the majority-rule procedure.

I. Introduction

THIS PAPER presents the results of an experiment to estimate the probability of error using unlabeled samples. Two procedures, along with the test results of each, are presented. The first procedure estimates the probability of error analytically, using the *a posteriori* density function. (The analytical estimate is shown to be unbiased.) The second labels fields (for use in estimating the probability of error) simply by noting the class into which most of the field picture elements (pixels) were classified by the classifier (called the *majority*