International Journal of Computer and Information Sciences, Vol. 6, No. 4, 1977

FEAST: Feature Evaluation and Selection Technique for Deployment in Unsupervised Nonparametric Environments[1]

Belur V. Dasarathy[2]

Received September 1975; revised February 1976

The problem of feature selection in a totally unsupervised, distribution free environment being conceptually ill-defined, the problem has been studied in an artifically evolved pseudosupervised environment. The evolution of such an environment is achieved by formulating a unified approach to the twin problems of feature selection and unsupervised learning. The solution of the latter problem leads to the pseudosupervised environment in which the features are evaluated by employing a multistate-choice automaton model as the feature selector. The methodology developed here is intended to be deployed in conjunction with any one of the numerous recursive schemes of clustering in which the crudely formed initial clusters are refined in a recursive fashion by successively determining the centroids of the different clusters and reallocating the samples to the clusters defined by these centroids. This allocation is carried out on the basis of distance measures (Euclidean or modifications thereof) and is in parallel progress with the feature-evaluation process. The clusters, as formulated at each stage of the recursive process, provide the pseudosupervised environment for the feature selector. The track record of the automaton in terms of probabilities of penalized action provides a measure of the efficiency of the different feature subsets in the unsupervised environment.

KEY WORDS: Feature selection in nonparametric unsupervised environments; simultaneous learning and feature selection; multistate-choice automaton for expedient learning in random environment.

[1] An earlier version of this paper was presented at the Fifth Annual Symposium on Automatic Imagery Pattern Recognition, University of Maryland, College Park, Maryland, April 1975.

[2] M & S Computing Inc., Huntsville, Alabama.

1. INTRODUCTION

The problem of feature selection in supervised environments, in which the distributions underlying the pattern classes are known, has been studied in depth through a variety of theoretically well-established approaches[1,2] such as those based on divergence, Bhattacharya distance, and similar concepts. On the other hand, feature selection in supervised but distribution free environments has also been studied in fair detail through nonparametric approaches[2,3] evolved from sound physical concepts such as interclass and intraclass distances as a measure of separability of classes. The problem of feature selection in unsupervised or imperfectly supervised environments is conceptually not well defined, although some attempts[4] have been made under the assumption that the probabilistic description of the different classes is available and that samples are available at least with imperfect labels. An alternative approach[5] that has been proposed is that of creating a pseudosupervised environment through a concurrent solution of the unsupervised learning problem, wherein the feature selection can be successfully carried out as before. This simultaneous unsupervised learning is achieved therein by updating the parameters of the distributions assumed to be known *a priori* in form, using the probabilistic,[6] the imperfect,[7] or the unfamiliar teacher scheme.[8] In a totally unsupervised and distribution free environment, neither a parametric approach, which calls for a knowledge of the distributions underlying the pattern classes, nor a nonparametric measure, which requires a labeled training samples set, can be construed for evaluating the effectiveness of a set of features. Therefore, a significantly different and innovative approach is necessary for tracking this problem of feature selection in an unsupervised, nonparametric environment. Such an approach is detailed in the sequel.

2. PROPOSED FEATURE-SELECTION TECHNIQUE

It is clear from Section 1 that this feature-selection problem can be defined in unambiguous terms only by creating a pseudo supervised environment and visualizing the feature-selection problem in that environment. This, in effect, calls for a unified approach to the twin problems of feature selection and unsupervised learning. Such an approach would tackle the two problems concurrently, with the solution of the unsupervised learning problem providing the pseudosupervised environment needed for solving the feature-selection problem. This pseudosupervised environment is continuously redefined as the learning progresses, thereby enhancing the reliability of the feature selector. This approach, although similar in framework to the one proposed earlier[5] for feature selection in parametrically

defined but unsupervised environments, differs significantly in the manner in which the pseudosupervised environment is derived. The probabilistic descriptions of the classes being unavailable, neither the probabilistic teacher scheme[6] nor its modifications[7,8] can be applied to obtain reliable labels for the given training sample set of unknown or unreliable classification. In view of this relatively less tractable environment, recourse is taken to the alternative approach of viewing the unsupervised learning problem as one of clustering the given set of unlabeled samples. The recursive process of homogenizing the initially formulated clusters, which is common to many clustering techniques,[1,9,10] provides the framework for continuously updating the labels of the sample set, thereby refining the pseudosupervised environment in which the feature selection can be carried out.

Here, the task of feature selection is viewed as one of evaluating the effectiveness of the different feature subsets and, in principle, can be carried out by any of the nonparametric approaches suited for operation in a supervised environment. However, in view of the fact that the environment is not a truly supervised one, but tends toward it gradually as the recursive clustering process progresses, it is observed that a feature-selection scheme that evaluates the features over a large span of operation in a statistical sense is most suited for integration with such a clustering process. The multistate-choice automaton model,[11] proposed recently for expedient learning in a random environment, has this property and is therefore employed here as the feature selector. This multistate-choice automaton model is a modification of an earlier finite automaton model proposed by Fu and Li[12] for learning in a stationary random environment and shown[13] to be applicable as a feature selector in a supervised environment. The new model[11] can change from a given state to any one of the other states under penalty, i.e., can choose any one of the possible actions whenever a penalty is received by the automaton, unlike the earlier model,[10] in which penalty dictated the automaton to go only to the next state. In the context of feature selection, this apparently small modification becomes conceptually significant, as the arbitrary order of identifying the different feature subsets with the different actions of the automaton no longer has any bearing on the outcome of the experiment. Thus, the feature-selection process becomes independent, as it should, of the arbitrary initial ordering of the feature subsets. Such ordering is necessary for associating the subsets with the different possible actions of the automaton.

Now consider the multistate-choice of automaton model[11] $M_{r,K}$ with K actions, each corresponding to r states. Here, it is to be noted that, for $K = 2$, this $M_{r,2}$ model will effectively reduce to the $A_{r,K}$ $(K = 2)$ model proposed earlier by Fu and Li,[12] both the models being conceptually equivalent in view of the fact that under penalty the automaton $M_{r,2}$ also

goes to the next state only (the only other available state). The $M_{r,K}$ model, in general, changes state under penalty input according to the state transition diagram (Fig. 2 of ref. 11). Because the goal of the automaton is to minimize the mathematical expectation of the penalty received by it, the expedient behavior of this learning automaton model can easily be established. This is not presented here, however, since it is available elsewhere in the literature.[11]

Here, each action k of the automaton is arbitrarily associated with a particular feature subset f_k and, depending on the number of feature subsets (say K) being evaluated, the automaton is selected to have K actions. Setting $r = 1$, for simplicity in presentation, the automaton model is considered in the form $M_{1,K}$. Under nonpenalty ($y = 0$), the automaton stays in its current state k, while under penalty ($y = 1$), the automaton takes a transition into one of the other states 1,..., i,..., K ($i \neq k$) according to the state transition diagram (Fig. 2 of ref. 11). The actual state it enters is decided by a random number generator with uniform distribution.

The feature-selection scheme shown in the figure can now be portrayed through the algorithm underlying the scheme.

Algorithm. Let the automaton $M_{1,K}$ be presently in state k.

1. As a new sample X^{j+1} with its pseudolabel L_{j+1} derived by the clustering scheme[10] is input to the system, derive the label λ_{j+1} by allocating the sample X^{j+1} to the cluster nearest to it (the nearness to a cluster can be based on the distance to its prototype such as the centroid or nearest neighbor among a set of prototypes of each cluster or other suitable concept). This distance is measured in the subspace defined by f_k^{j+1}, the subset of features corresponding to the present state of the automaton.

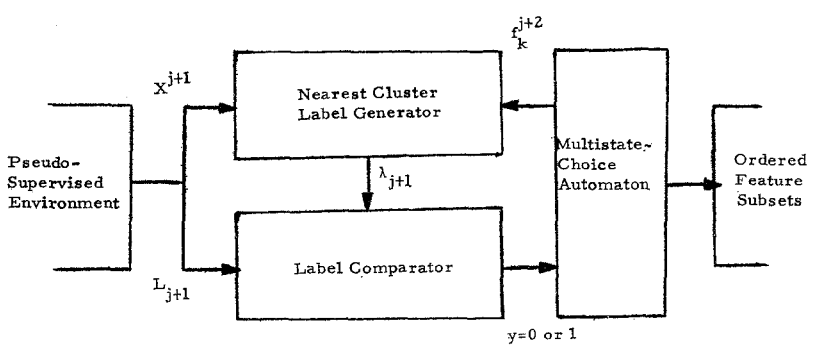

Fig. 1. Schematic representation of feature evaluation and selection technique (FEAST).

2. Compare the labels L_{j+1} and λ_{j+1}, and set y according to the following rule:

$$L_{j+1} = \lambda_{j+1} \Rightarrow y = 0 \text{ (nonpenalty)}$$
$$L_{j+1} \neq \lambda_{j+1} \Rightarrow y = 1 \text{ (penalty)}$$

3. Under nonpenalty, i.e., $y = 0$, the automaton remains in the present state k; while under penalty $y = 1$, the automaton jumps to one of the other states, the actual state at this stage being determined by a newly generated, uniformly distributed random number between 0 and $K - 1$. The corresponding output f_k^{j+2} of the automaton dictates the feature subset to be used in processing the next sample.

4. Go back to step 1 to receive the next input sample X^{j+2}. The probability π_k of receiving a penalty for deploying a particular feature subset f_k is estimated as

$$\pi_k = \frac{\alpha_k}{\beta_k}$$

where β_k is the frequency of the action f_k (i.e., the number of times the feature subset f_k was used) and α_k is the frequency of penalized action f_k (i.e., the number of times a penalty was received by the automaton for using this feature subset f_k).

A track record of the automaton, in terms of the number of times each feature subset was used along with the corresponding number of penalties received by the automaton for using these subsets, is maintained, and this is used to estimate $\pi_k : k = 1,..., K$. These values of $\pi_k : k = 1,..., K$ are then treated as measures of the effectiveness of the feature subsets and used to order the different subsets accordingly.

3. COMPUTATIONAL ASPECTS

The feasibility of the approach was investigated using the C-1 flightline data of the Laboratory for Applications of Remote Sensing (LARS) of Purdue University.[14] These data are sensed from an aircraft through a 12-channel multispectral scanner. Well-identified training samples of this data set were used in these experiments by withholding the known labels of these samples to simulate the unsupervised environments. Here, these labeled samples were used to help the *a posteriori* evaluation of the results of clustering and feature selection. The very first problem encountered in the implementation was that the proposed scheme entailed setting up an automaton $M_{1,K}$, where K is exceedingly large if one has to evaluate all possible feature subsets (of sizes 1 through 12). For this case, $K = 2^{12} - 1$ and, in general, $K = 2^N - 1$, where N is the number of individual features

defining the data set. Even if only subsets of a particular prespecified dimension, say n ($< N$), are to be assessed, one would still need

$$\binom{N}{n} = \frac{N!}{(N-n)!\,n!}$$

states for the automaton. For $n \simeq N/2$, this signifies a very large number of states. This in turn calls for a prohibitively large number of samples to be processed by the feature selector before the statistics derived by the technique can be considered reliable and sufficient. To overcome this inadequacy of the method in the processing of large dimensional data sets, a suboptimal sequential procedure can be constructed.

Such a procedure involves significantly less computational load without sacrificing the essential ingredients of the scheme. The scheme would call for at most N states for implementation of the automaton $M_{1,K}$, i.e., $K \leqslant N$, at any stage of the sequential procedure. In practice, the procedure can be visualized in two ways:

1. Determine the single best feature out of the given N features at the first stage, combine this best feature with each of the remaining $(N-1)$ features to determine the best subset of two features at the next stage, and continue in the same manner until the best feature-subset of requisite dimension n is derived.

2. Alternatively, determine the single worst feature, i.e., evaluate the best feature subset of $(N-1)$ dimension out of the possible N such subsets (derived by dropping out one or the other of the N features) at the first stage, similarly determine the best subset of $(N-2)$ dimension at the next stage, and continue until the best subset of the required dimensionality is obtained.

Choice of one of these two procedures is dictated by the value of n relative to N, as that decides the computational expense involved in the sequential process. Conceptually, of course, the latter is more satisfying as reliance is at no time placed on the performance of a single feature. However if n is closer to unity, i.e., if a very small set is sought, the former approach is better suited. If n is closer to N, then the latter procedure is both conceptually and computationally more desirable in view of the lesser number of steps in the sequential process. But, it is to be noted that the computation needed at each step in the latter approach is higher because the distances are measured in a correspondingly higher-dimensional space. Thus, strictly from a computational point of view, the two approaches balance out, not at $n \simeq N/2$, but at a value of n much closer to N. Depending on whether

$$\sum_{i=1}^{n} \begin{matrix} < \\ > \end{matrix} \sum_{i=n}^{N-1} i \quad \text{or}$$

i.e., according to whether

$$n^2 \begin{matrix} < \\ \text{or} \\ > \end{matrix} \frac{N(N-1)}{2}$$

either the former or latter approach is to be adopted. Of course, one could visualize a far simpler implementation, namely, ordering the features by evaluating all the individual features simultaneously in one stage and selecting the n best features directly out of the ordered set of N features. This is considered unsuitable because the effect of cross-correlation between features is not taken into account at all by this overly simplified procedure. For example, if two features are individually determined as good in discriminating between the same two classes, the combination of these two features may not necessarily be superior to the combination of using the better of these features with another feature that is good in discriminating a third class. This is particularly significant in a multiclass environment. This, therefore, justifies the higher computational expense of the sequential procedure outlined above.

4. TEST RESULTS AND CONCLUDING REMARKS

This sequential procedure was coded and tested using the nine-class 12-dimensional training data set (of beans, corn, oats, rye, alfalfa, soil, red clover, wheat I, and wheat II) of Purdue LARS C-1 flightline data set. The labels of the input data set were withheld to simulate the unsupervised environment. The clustering resulted in essentially 10 clusters with a good one-to-one correspondence between the clusters as evolved and the known pattern classes except for the case of red clover, which separated into two separate clusters. A close scrutiny of the red clover samples reveals a rather large spread along features 11 and 12 (which were discerned as the most significant features) compared with the variations within other classes and even between other classes. This, of course, is not a serious discrepancy, as subclusters can always be recombined at a later stage. This multiclass environment (rather than a simple two-class one) was used for testing because it represents a more realistic unsupervised envrionment in which one has no knowledge of the actual number of classes. Of course, this makes the evaluation of the feature-selection results a little more complex, as most of the other published results[5,13,14] of feature selection on this data set relate to two class environments. Even those[3] that relate to multiclass supervised environments tackle the problem as a set of sequential or batch two-class problems rather than as a concurrent multiclass problem. In spite of these differences, the feature ordering obtained here (Table I) compares favorably (albeit not identically) with that derived elsewhere through supervised[3,13,14] or unsupervised

Table I. Feature Ordering Derived by Processing C-1 Flightline Training Data Set Through FEAST

Feature Order
11
12
5
2
8
10
6
9
1
3
4
7

methods.[5] Of course, the discrepancies result because the feature subset optimal for a particular pair of classes is not necessarily the best for the total multiclass problem environment and because the clusters resulting from the clustering process cannot be exactly matched with the external class distinctions. The feature ordering thus obtained can be employed to define the best feature subset of any dimension, say n, by using the first n features of the ordered set.

It is therefore believed that this feature evaluation and selection technique (FEAST) can be reliably employed for feature selection in unsupervised nonparametric environments, requisitioning the suboptimal sequential procedures whenever the dimensionality of the data set is relatively large.

REFERENCES

1. K. Fukunaga, *Introduction to Statistical Pattern Recognition* (Academic Press, New York, 1972).
2. *IEEE Trans. Comput.*, "Special Issue on Feature Extraction and Selection in Pattern Recognition," **C-20**:965 (1971).
3. B. V. Dasarathy, "An integrated nonparametric sequential approach to multiclass pattern classification, *Int. J. Syst. Sci.* **4**:449 (1973).
4. C. Babu, "On the application of divergence for the extraction of features from imperfectly labeled patterns," *IEEE Trans. Syst. Man Cybern.* **SMC-2**:290 (1972).
5. A. L. Lakshminarasimhan AND B. V. Dasarathy, "A unified approach to feature selection and learning in unsupervised environment," *IEEE Trans. Comput.* **C-24**:948 (1975).
6. A. K. Agrawala, "Learning with a probabilistic teacher," *IEEE Trans. Inf. Theory* **IT-16**:373 (1970).

7. K. Shanmugam, "A parametric procedure for learning with an imperfect teacher," *IEEE Trans. Inf. Theory* **IT-18**:300 (1972).
8. B. V. Dasarathy and A. L. Lakshminarasimhan, "Sequential learning employing unfamiliar teacher hypothesis (SLEUTH) with concurrent estimation of both the parameters and teacher characteristics," *Int. J. Comput. Inf. Sci.* **5**:1 (1976).
9. G. H. Ball and D. J. Hall, "ISODATA, An Iterative Method of Multivariate Analysis and Pattern Classification," *International Communication Conference*, Philadelphia (1966).
10. B. V. Dasarathy, "An innovative clustering technique for unsupervised learning in the context of remotely sensed earth resources data analysis," *Int. J. Syst. Sci.* **6**:23 (1975).
11. A. L. Lakshminarasimhan and B. V. Dasarathy, "Multi-state-choice automata model for expedient learning in random environment," *Inf. Sci.* **9**:91 (1975).
12. K. S. Fu and T. J. Li, "Formulation of learning automata and automata games," *Inf. Sci.* **1**:237 (1969).
13. K. S. Fu, P. E. J. Min, and T. J. Li, "Feature selection in pattern recognition," *IEEE Trans. Syst. Sci. Cybern.* **SSC-6**:33 (1970).
14. K. S. Fu, D. A. Landgrebe, and T. L. Phillips," Information processing of remotely sensed agricultural data," *Proc. IEEE* **57**:639 (1969).