FEATURE SELECTION ALGORITHMS USING NON-REDUNDANT THRESHOLDED MEASURES

by

Edward G. Fisher*
Allen R. Hanson†
Edward M. Riseman*

COINS Technical Report 74C-9

December 1974

This research was supported by the Office of Naval Research
under Grant N00014-67-A-0230-0007

*Computer and Information Science
University of Massachusetts
Amherst, MA 01002

†School of Language and Communication
Hampshire College
Amherst, MA 01002

I. INTRODUCTION

This paper describes a new feature selection algorithm, called the threshold selection algorithm, which is an extension of the sequential procedure studied by Estes [1]. The method allows the selection of a subset of features from a predefined pool of features and takes into account both global and local characteristics of the subset selected. The threshold selection algorithm provides a set of parameters which are dynamically variable by the system employing the algorithm or which may be preset by the designer of the system. It allows a tremendous degree of flexibility in tuning the feature selection phase to the recognition problem under consideration.

Briefly, associated with each class pair in pattern space is a threshold which provides information to the selection algorithm specifying the amount of effort to be expended in selecting features which separate the class pair. In this way, more effort may be expended for those class pairs which are particularly troublesome to the classifier and following systems (e.g., a contextual postprocessor). The algorithm maintains a global perspective by considering all class pairs which have not yet received the effort indicated by the thresholds while at the same time considering more heavily those class pairs which are further from their thresholds.

A variable emphasis may be placed on different class pairs according to the wishes of the system or the designer. For example, in the case of alphabetic recognition problems, it may be desirable to select more powerful features for discrimination between topologically similar characters (U and V, for example). On the other hand, since some class pairs <u>are</u> topologically similar, effective features for discriminating between these pairs may not be present in the pool. Consequently, the selection algorithm should not concen-

trate unduly on a-priori troublesome class pairs; we might be content with concentrating on the easier cases, allowing the remainder of the system (in particular a contextual postprocessor [2,3]) to handle the difficult classes. Similarly, if one knew which classifier errors are most troublesome to the contextual postprocessor, the threshold associated with each of these pairs may be raised in an effort to reduce the number of errors from the classifier which are not correctable by the contextual subsystem. For example, if the classifier substitutes an E for an A, and the contextual subsystem is frequently unable to correct this particular error, the class pair AE should receive more attention.

The feature selection problem which we consider here is the selection of a subset of N features from an available pool of $\mathcal{R}$ features, $\mathcal{P} = (F_1, \ldots, F_k)$. That is, $\mathcal{P}$ contains features thought to be useful for recognition in a particular problem domain and we wish to select that subset which is the "best," perhaps yielding the lowest system error rate, or the least computationally expensive, etc. In most cases, the number of possible subsets of size N from a set of features of size k is too large to permit exhaustive evaluation of each subset. For example, if $k = 200$ and $N = 30$, as it is in some of the experiments described later, the number of subsets which must be evaluated is on the order of 4.1×10^{36}. Therefore, we must relax the constraints and accept suboptimal feature subsets.

Various methods have been proposed for selecting a "good" subset of features [1,4-8]. These methods generally select a set of features which maximizes the inter-class distances in the transformed space based upon a training set. Multi-class problems are normally treated as separate pairwise

two-class problems [8]. Furthermore, simplifying assumptions concerning independence of the feature measurements are usually made to facilitate the selection process. Once this assumption is made, the distances associated with individual features comprising a subset may be added to yield a measure of the overall effectiveness of the entire subset. In actual practice, this condition is rarely met and the result is a set of features which may be highly redundant and whose quality is uncertain. Since the classifier is also very often designed with an assumption of independence, the problem is compounded and the overall system error rate is increased.

II. DISCRIMINANT MEASURES

Many distance measures have been proposed in the literature for feature evaluation; most of these are summarized in [8,9]. We have chosen a distance measure originally proposed by Bakis, Herbst, and Nagy [10] because of its computational simplicity. However, the general conclusions reached are not dependent upon the choice of measure. It is our contention that very simple measures are suitable if the remainder of the system is designed properly.

Very briefly, for each feature $F_j (1 \leq j \leq k)$ in the pool P we compute the sample mean and variance of F_j when applied to each character class; thus μ_α^j and σ_α^j are, respectively, the mean and variance of the measurement F_j when applied to the sample characters labelled α in the training set; α ranges over the entire set of classes, here A - Z. A measure of the utility of F_j in discriminating between the class pair $\alpha\beta$ is given by $f_{\alpha\beta}^j$ as defined in (1).

$$f^j_{\alpha\beta} = \frac{(\mu^j_\alpha - \mu^j_\beta)^2}{\sigma^j_\alpha + \sigma^j_\beta} \qquad (1)$$

The j^{th} feature is thus characterized by a vector of values $f^j_{\alpha\beta}$, where $\alpha,\beta \in \{A,B,\ldots,Z\}$, $1 \leq j \leq k$ and $\alpha \neq \beta$.

Intuitively, $f^j_{\alpha\beta}$ is an inverse measure of the overlap of the class conditional probability density functions since it is an increasing function of the squared difference between the class conditional means and a decreasing function of the sum of the class conditional variances of F_j. One disadvantage is that $f^j_{\alpha\beta}$ is most reliable when the class-conditional probability densities are Gaussian [9]. Multimodality and skewness are not adequately represented by means and variances; therefore, these characteristics are not accurately described by the utility measure.

III. EXPERIMENTAL DESIGN

The pool of features $\mathcal{F} = (F_1, \ldots, F_k)$, for the experiments described in the following sections was of size 200, consisting of 52 randomly designed n-tuples, 22 hand-picked n-tuples, 45 high information n-tuples, 8 topological measurements, and 25 windows. A more detailed description of the feature pool may be found in Appendix A. The goal of the experiments described below was to select from this pool a subset consisting of approximately 20 to 50 features. Virtually no effort was made to hand-tailor the pool to ensure a set of good features.

The data upon which the experiments are based were derived from the IEEE data set consisting of Munson's multi-author hand-printed characters [11]. Three sets of alphabetic characters from each of 49 authors resulted in a set of 3,722 characters. The feature subsets were selected and tested

on the basis of statistics gathered from a training set of 98 alphabets (two per author) while the remaining 49 alphabets constituted the test set. The characters from this test set were used to create 6-letter English words which formed the actual input to the system (character by character). The classifier employed throughout the experiments is the simplest Bayesian classifier, assuming independence of measurements and no underlying parametric distributions; similar results were obtained assuming normal, distributed measurements with equal covariance matrices.

Error rates quoted throughout the remainder of this paper were computed by assuming that the characters of the test set occurred in proportion to their occurrence in English text [12] since the input to the system is a sequence of characters which are ultimately viewed as whole words. These error rates were estimated as in (2).

$$E = \sum_{i=1}^{26} P_e(C_i)P_i \qquad (2)$$

$P_e(C_i)$ is the expectation that the classifier makes an error on a test sample from the i^{th} class and P_i is the a-priori probability of occurrence of class i. In addition, error rates were also computed assuming that the characters occur uniformly in the pattern space. While these results are not reported here, they are consistent with those of Hussain, Toussaint, and Donaldson [13] in that textual error rate estimates are almost always lower than those estimates obtained by assuming equi-probable characters. We should emphasize that the error rates reported here do not reflect the substantial reduction achieved after contextual postprocessing [3]. They are based solely on the Bayesian classifier statistics.

IV. SELECTION METHODS

The feature selection schemes investigated here are multi-step processes. We distinguish between two general methods for subset generation: sequential selection and sequential rejection. For sequential selection, the subset to be chosen is initially empty; features are selected from the pool according to the criteria discussed below. In feature rejection, the subset to be chosen is assumed to consist initially of the entire pool; features are rejected from the subset until the desired size is reached. We now compare the results of feature subset selection using random selection, sequential selection and sequential rejection.

Random Selection

The simplest algorithm would be to simply use all available features, in this case 200. This method is not to be seriously considered for computational reasons; in general, the size of the pool can exceed available resources and a method is needed to reduce the set. The resultant error rate is 8.2%. This error rate is not to be construed as any sort of limit since there exist many dependencies among the features in the pool.

The next algorithm to be considered simply selects features from the pool at random until the requisite number of features has been obtained. Ten different subsets ranging in size from 10 features to 100 features were selected and tested. The results are summarized in Table 1. For the results quoted in this table, each subset of size n is a proper subset of features of size $n + 10$.

In general, the results quoted for the different selection algorithms must be interpreted with care. For example, the apparent limit in Table 1 may be as much due to the fact that there are only 200 features in the pool

as the presence of any real "limit" in the performance of the system. For feature subsets containing a reasonably large fraction of the pool, any selection algorithm which assumes independence will necessarily

Set #	n_f: number of features									
	10	20	30	40	50	60	70	80	90	100
1	53.0	27.6	24.4	20.7	16.5	15.9	12.5	12.2	11.6	10.9
2	55.4	35.1	25.8	17.4	17.8	16.7	15.2	13.0	12.0	11.0
3	55.7	37.4	23.0	20.1	17.7	14.15	14.0	12.4	12.3	11.5
Average	54.7	33.4	24.4	19.2	17.3	15.7	13.9	12.5	12.0	11.1

Table 1. Error rates (%) randomly selected feature sets.

choose redundant features, assuming no effort is made to pre-condition the feature pool.

Sequential Selection and Rejection

We first treat the case of sequential selection. At each step there is a set of n features; we then select the best additional feature given those first n features. That is, given what is known about the discriminating ability of the first n features,[1] it is desirable to select a feature which, when added to that subset, will in some way improve its discriminating ability. The difference between the various algorithms to be described is the method of determining the "best" additional feature. The process terminates when enough features have been selected or a good enough subset has been obtained.

[1]The information content of these n features collectively is less than or equal to the sum of the information measures taken for the n features independently.

In notational terms, we say that at the (n+1)-th step in the selection process, we have a subset $\mathcal{f}_n \subset \mathcal{P}_n$ where $\mathcal{f}_n = \{F_{i_1}, F_{i_2}, \ldots, F_{i_n}\}$ i.e., $\mathcal{f}_n$ contains n features from the pool $\mathcal{P}$. Then we must select a feature $F_{i_{n+1}}$ so that $\mathcal{f}_{n+1} = \mathcal{f}_n \cup \{F_{i_{n+1}}\}$ is a better set of features than $\mathcal{f}_n$. The criterion for selection of a feature F_j is a function of the set of class pair utility measures assigned to the feature. Recall that F_j has an associated set of utility measures $\{f^j_{AB}, f^j_{AC}, \ldots, f^j_{YZ}\}$. The sequential procedures each utilize a set of accumulators $\{a_{AB}, a_{AC}, \ldots, a_{YZ}\}$, such that, at the (n+1)-th step, each accumulator is defined as in (3).

$$a_{\alpha\beta} = \sum_{F_i \in \mathcal{f}_n} f^i_{\alpha\beta} \tag{3}$$

Thus, the accumulator $a_{\alpha\beta}$ provides a measure of the total amount of separation or discriminating ability between α and β in $\mathcal{f}_n$; and the entire set of accumulators describes the worth of the set $\mathcal{f}_n$ over all class pairs.

For the sequential selection procedure, the feature set $\mathcal{f}_1$ may be arbitrarily selected; for the results quoted here, that feature F_j is chosen such that

$$\sum_{\substack{\alpha,\beta \\ \alpha \neq \beta}} f^j_{\alpha\beta}$$

is a maximum--but $\mathcal{f}_1$ could be selected by any technique and may be defined as containing more than one feature. Then, at each subsequent step, the accumulators are computed and the smallest accumulator, $a_{\alpha\beta}$, is determined. The feature selected is that feature $F_j \in \mathcal{P} - \mathcal{f}_n$ for which $f^j_{\alpha\beta}$ is a maximum. Intuitively, this means that we determine that pair of classes for which the feature set has the least discriminating ability; a feature is then selected from the pool which does more for that class pair than does any of the other

features left in the pool.

Bakis, *et al.*, [10] used a sequential rejection scheme in which the initial set of features contained the entire pool and the accumulators were defined as

$$a_{\alpha\beta} = \sum_{F_i \epsilon \mathcal{F}^{\alpha\beta}} f^i \tag{4}$$

At each step the smallest $a_{\alpha\beta}$ is determined and that feature F_j for which $f^j_{\alpha\beta}$ is a minimum is deleted from $\mathcal{F}_1$ to form $\mathcal{F}_2$; that is, a search is made for that pair of classes which is most poorly discriminated (according to our measure) and the feature which provides the least discrimination for that pair of classes is deleted. The process continues until a set of features $\mathcal{F}_n$ is obtained of the desired size.

In each of these two techniques, a feature is selected or deleted on the basis of its performance in discriminating between the members of a particular pair of classes; thus, a feature which could always discriminate between O and V might not be selected (or might be deleted) because the algorithm spends a great deal of effort searching for discriminators between O and D. Note that this only implies that $a_{OV} > a_{OD}$ but says nothing about their relative magnitudes. These algorithms are unsatisfactory for problems with a great number of classes because they disregard the global capabilities of features in favor of a single point at which most of the features are weak. In their attempt to give best possible discrimination between the worst class pairs, these algorithms may give nearly random discrimination for most other decisions.

Generalized Sequential Selection and Rejection

A generalization of these techniques which suggests itself is that we select a parameter m and at each step, instead of optimizing our discrimin-

ation on the single worst class pair, we optimize on the m worst class pairs. This would be expected to yield a better set of features since it increases the size of the local area in the class-pair space involved in the selection (rejection) decision.

Table 2 presents the results of experiments with these algorithms for sets of ten to one hundred features. Several conclusions may be drawn from these data. Sequential selection yields better results when $m = 50$ than when $m = 1$. Sequential rejection is not better than a random selection of features (Table 1) when $m = 1$ but achieves significant improvements when $m = 10$ and $m = 50$. The sequential selection scheme is significantly better than sequential rejection until $m = 50$ at which point there is little difference between them. In the range of thirty to sixty features per set, we have what are apparently the best results for the two techniques at $m = 50$.

m \ n_f: Number of features	10	20	30	40	50	60	70	80	90	100
	Sequential Selection									
1	31.9	20.0	15.0	14.1	14.7	11.6	11.5	10.5	10.5	9.2
10	22.7	15.4	15.7	12.6	11.4	11.1	9.6	10.3	9.7	9.2
25	24.3	17.4	16.3	12.9	11.1	11.1	11.3	10.7	10.5	10.3
50	25.1	18.8	14.2	11.5	10.5	10.6	10.2	10.6	9.6	10.3
100	25.1	17.9	15.1	12.8	12.1	10.5	11.6	10.4	9.4	9.8
	Sequential Rejection									
1	53.6	38.3	27.0	22.4	19.6	16.6	17.3	13.0	12.4	10.9
10	39.2	22.5	19.7	16.6	13.5	11.5	10.7	9.9	9.9	9.3
25	31.3	20.4	15.7	14.6	13.3	11.7	10.4	9.6	9.6	8.7
50	34.9	18.6	14.2	11.6	10.9	10.7	9.7	10.0	9.7	10.2
100	30.1	16.7	14.3	12.0	12.0	11.3	10.1	9.2	8.8	8.8

Table 2. Error rate estimates from feature subsets selected by the sequential selection and sequential rejection techniques.

Estes [1] (using different utility measures) concluded that under certain circumstances sequential selection and rejection lead to equivalent results and that sequential rejection was probably not worth the additional computation. Since our primary interest in the experiments reported here was for the cases in which the size of the selected subset was small compared to the size of the pool, say 20 to 40 features from 200, we concur with this observation. From our results, we might conjecture that the two methods yield similar results when the appropriate proportion of class pairs are considered for worst case analysis. Bakis, *et al.*, considered the case of hand-printed numerals in which there are 45 class pairs instead of the 325 of the alphabetic character problem; thus, $m = 1$ was probably very meaningful for their study but too small to be useful in ours. However, we expect that m in the range of 2 to 8 would have produced better results in their case.

V. THRESHOLD SELECTION ALGORITHMS

Even with the improvement obtained by considering several class pairs for worst case analysis, it seems that much of the discriminating ability of the selected feature sets is somewhat randomly distributed among the class pairs. A more globally oriented technique would pay attention first to the entire set of class pairs and then to some smaller set of those which seem to need additional discrimination. In the algorithm described here, a threshold is associated with each class pair. The threshold $\theta_{\alpha\beta}$ represents the total separation desired for the $\alpha\beta$ class pair; i.e., $\theta_{\alpha\beta}$ is the value which the accumulator $a_{\alpha\beta}$ is expected to achieve. At each step, the algorithm selects the feature which minimizes the global (total) difference between the thresholds and the accumulators. When an accumulator reaches its threshold,

no further improvement is sought for its class pair. The contribution c_j of feature F_j is computed as

$$c_j = \sum_{\substack{\alpha,\beta \\ \alpha \neq \beta}} k^j_{\alpha\beta} \tag{5}$$

where

$$k^j_{\alpha\beta} = \begin{cases} 0 & \text{if } a_{\alpha\beta} \geq \theta_{\alpha\beta} \\ f^j_{\alpha\beta} & \text{if } f^j_{\alpha\beta} \leq \theta_{\alpha\beta} - a_{\alpha\beta} \\ \theta_{\alpha\beta} - a_{\alpha\beta} & \text{if } f^j_{\alpha\beta} > \theta_{\alpha\beta} - a_{\alpha\beta} \end{cases}$$

Thus, in the initial steps the threshold selection procedure attempts to minimize the total difference between the thresholds and the accumulators. A global outlook is maintained since the algorithm considers all class pairs until their thresholds are reached. However, the accumulators often achieve or exceed their respective thresholds since there are many class pairs which are well discriminated. Consequently, in the later steps, the algorithm selects features in an attempt to minimize the threshold-accumulator differences for the remaining class pairs. Each class pair $\alpha\beta$ potentially affects the selection decision in proportion to the difference between $a_{\alpha\beta}$ and $\theta_{\alpha\beta}$.

A great deal of latitude is available in the definition of the thresholds. A simple technique would make all of the thresholds the same, implying an equal minimum discrimination is desired for all class pairs. On the other hand, it is often very difficult to provide equal discrimination between all class pairs; we can easily discriminate between T and O but we often confuse U and V, etc. Furthermore, in some problems it is more important for some class pairs to be well differentiated than it is for others. For

example, in a contextual system, it is more important that we not mistakenly classify an E as an I than an E as a Z since such substitutions produce more undetectable errors [2].

Thus, several alternative definitions are available. The thresholds could all be set at some arbitrary constant, or more generally, some constant times the number of features to be selected

$$\theta_{\alpha\beta} = c \cdot n_f \tag{6}$$

to allow a different threshold level for different size feature sets. To account for the fact that it probably is not possible to differentiate all class pairs equally well, each class pair threshold could be evaluated as a proportion of the pool's total discriminating ability for that class pair

$$\theta_{\alpha\beta} = g(n_f, \mathcal{J}_{\alpha\beta}) \quad \text{where} \quad \mathcal{J}_{\alpha\beta} = \sum_{F_j \varepsilon} f^j_{\alpha\beta} \tag{7}$$

or as some function of the mean and variance of the $\{f^j_{\alpha\beta}\}$

$$\theta_{\alpha\beta} = g(n_f, \mu_{\alpha\beta}, \sigma_{\alpha\beta}) \tag{8}$$

To account for our knowledge that we must provide better discrimination for some class pairs than for others, we can define

$$\theta_{\alpha\beta} = g(n_f, c_{\alpha\beta}) \tag{9}$$

where $c_{\alpha\beta}$ is a measure of the confusion or loss caused by classifying α as β and/or β as α.

Experiments were conducted using the threshold feature selection algorithm on our pool of features. The thresholds were computed as in (6) with the constant assuming the values $\frac{1}{3}$, $\frac{1}{2}$, and $\frac{2}{3}$. These values were chosen

because lower values resulted in each accumulator exceeding its threshold before the desired number of features were selected and higher values resulted in markedly poorer performance. The resulting error rate estimates are shown in Table 3 together with the number of accumulators which exceeded their thresholds at the termination of the feature selection process. The only clear point among these results is that the threshold selection algorithm does do significantly better than previous algorithms for sets of up to 50 features. For any given number of features, it seems to be necessary to apply the algorithm several times since it is not apparent that one can automatically select the proper value for the constant. However, use of (8) could allow the system to automatically set the range as a function of the actual values for the figure of merit.

	n_f: Number of features									
c	10	20	30	40	50	60	70	80	90	100
1/3	E: 22.8	15.0	13.0	11.1	10.9	10.4	10.9	10.1	8.8	8.3
	N: 317	322	319	317	316	315	314	313	310	308
1/2	E: 26.2	14.2	12.4	10.8	11.2	10.8	9.7	10.1	10.2	9.7
	N: 311	313	309	305	303	296	293	292	286	284
2/3	E: 23.3	15.2	10.9	11.3	10.9	10.4	9.9	9.7	9.7	9.1
	N: 299	296	295	289	286	280	275	276	263	258

Table 3. Error rate estimates obtained from feature sets selected by the threshold selection algorithm. E is the error rate estimate and N is the number of accumulators which exceeded their thresholds.

VI. THRESHOLD SELECTION WITH REDUNDANCY MEASURES

A drawback to each of the techniques discusses is that none of them prevents the selection of highly redundant features. However, normal procedure is to use a classifier assuming statistical independence among features; violation of this assumption will certainly degrade its performance. Although each feature selected might be very good in its own right, the total amount of information to be extracted about any particular character might be far less than expected.

Redundancy among features has been studied by Mucciardi and Gose [7], among others. A feature utility measure was defined as a weighted sum of the probability of error (POE_j), and the average correlation measure (s_j) of the feature with each of those previously selected. The (n+1)-th feature selected was that feature F_j for which c_j is a minimum:

$$c_j = w_1 (POE_j) + w_2 s_j \qquad (10)$$

$$\text{where } s_j = \frac{1}{n} \sum_{F_i \in f_n} \left| r_{ij} \right|$$

and r_{ij} is the Pearson correlation coefficient.

When the weights were adjusted for optimal recognition, the average correlation coefficient was nine times more important for selection than was the probability of error (i.e., $w_1 = .1$ and $w_2 = .9$).

The probability of error of Mucciardi and Gose was not used as a feature evaluation function since all of our features assume fewer than twenty-six values; most of the features are binary. While POE is a convenient utility measure, it does not provide a very good description of the features employed in this study.

Using the average correlation coefficient as a measure of redundancy for the threshold feature selection algorithm resulted in error rates which were consistently worse than the error rate obtained for the threshold feature selection algorithm without a redundancy measure. An average of correlation coefficients is not sufficiently sensitive to highly interrelated features. For example, suppose a feature is highly correlated to some already selected feature. However, if many of the other correlation coefficients are small, the average of these coefficients may result in a small average correlation and allow this feature to compare favorably with other features which are only slightly correlated with all of the selected features. This feature should not be selected; a feature is highly redundant if it is highly correlated with <u>any</u> feature in the set. This is not to say that small correlations do not accumulate to form high multiple correlations, but there are so *many* coefficients that many of them are spurious and must be systematically ignored.

All of the feature selection algorithms considered may be modified to incorporate a measure of redundancy. At the (n+1)-th step, the proportion of information represented by F_j which is not already provided by the features selected can be approximated by t_j, computed as in

$$t_j = 1 - r_j$$

$$r_j = \max_{F_i \varepsilon \mathcal{F}_n} \left| r_{ij} \right| \qquad (11)$$

where r_{ij} is the Pearson correlation coefficient, $0 \leq |r_{ij}| \leq 1$.

The contribution of F_j to the discrimination may be found by computing a new utility measure $f'^{j}_{\alpha\beta}$ in (12).

$$f'^{j}_{\alpha\beta} = f^{j}_{\alpha\beta}\, t_j \quad \text{for each} \quad \alpha,\beta. \tag{12}$$

Intuitively, $f'^{j}_{\alpha\beta}$ is the non-redundant information provided by F_j in distinguishing α from β.

Table 4 presents classifier error rate estimates for the threshold selection algorithm when it was modified to include the redundancy measure described above. By comparing with Table 3, it is apparent that incorporating the redundancy measure into the threshold selection algorithm results in a better set of error rates than that obtained without such a measure. Clearly, these error rate estimates are consistently better than those obtained from any of the earlier techniques.

	n_f: Number of features									
c	10	20	30	40	50	60	70	80	90	100
1/3	21.8	14.0	11.0	11.3	9.3	10.4	9.5	9.3	8.6	8.7
1/2	22.5	13.4	11.6	10.3	9.4	9.3	8.9	9.5	8.8	8.7
2/3	22.5	14.1	12.1	10.8	9.9	9.0	8.8	8.3	8.7	8.0

Table 4. Error rate estimates from feature sub-sets obtained from the threshold selection algorithm with the redundancy measure of equation (11).

When comparing the results of the several algorithms, one is led to wonder about the small differences obtained from the large sets of features. This phenomenon might be explained by considering the fact that the feature pool has only two hundred features. Sets of fifty to one hundred features represent a large proportion of the features in the pool and have a fairly large intersection with each other. Thus, if the pool were much larger, one might be able to compare the algorithms' performance on larger feature sets. We believe the threshold selection algorithm with correlation is better than any of the

other algorithms; we suspect that the reason that it did not do a great deal better on the larger feature sets is because of the small size of the feature pool. We expect that any small but consistent differences would be accentuated with a larger feature pool.

VII. SUMMARY AND CONCLUSIONS

A standard Bayesian classification system was applied to the IEEE data set of Munson's multi-author hand-printed characters to compare empirically the following feature selection algorithms:

1. random selection
2. generalized sequential selection
3. generalized sequential rejection
4. threshold selection

and 5. threshold selection with a redundancy measure.

Random selection has been included for completeness; in a general sense, it provides a basis of comparison for the remaining algorithms, although such comparisons must be made with care. It is difficult to attach a meaningful significance measure to the results reported here, hence comparisons between single error rate estimates should be made with care. On the other hand, since we are comparing the results of competing algorithms, if one algorithm is consistently better than another over wide variations in parameters, we tend to believe that this did not occur by chance alone. We would tend to have faith in the superiority of one algorithm, regardless of the lack of significant differences among individual results.

The feature selection algorithms considered here depend upon the existence of a measure which is an estimate of the effectiveness of a particular

feature. Usually, such a measure is an estimate of the separation of the class-conditional probability densities in the feature space. This quantity is usually related to the probability of error of the classifier [9]. Bakis, Herbst, and Nagy [10] defined a measure related to the Bhattacharyya distance [9] and obtained good results in the case of hand-printed numerals, utilizing the sequential rejection algorithm to select a feature set from a pool of features; this measure was used in each of the algorithms discussed here (except random selection). Sequential rejection deletes from the pool that feature which contributes least to that class pair with the smallest total measurement until the desired number of features remains. The sequential selection algorithm is similar in that features are chosen sequentially to aid the worst case. Both of these are esentially worst-case designs; features which normally would provide adequate separation are discarded (or never added) solely on the basis of their local inferiority.

Both sequential rejection and selection can be generalized by basing the decision for rejection or selection on the m worst class pairs. For the 325 class pair problem considered, performance of the algorithms improved when m was increased from 1 to 25 or 50 (Table 2). The generalization provides a more global decision mechanism. Furthermore, from the results quoted previously, sequential rejection is marginally inferior to sequential selection for small sets of features (subsets in the range of 30 to 60 features out of 200). However, rejection is computationally more expensive for subset sizes less than one-half of the pool size. Under these conditions, we see no reason for using rejection techniques.

In an effort to provide a more rational decision for feature selection, the threshold selection algorithm is proposed. This selection algorithm defines a threshold associated with each class pair ($\theta_{\alpha\beta}$). This value represents the

required or desired separation that should be achieved between class α and class β by the selected set of features. Those class pairs which satisfy their threshold no longer influence the subsequent selection process. The feature selected next is that feature which moves the remaining class pairs the most towards their thresholds. Features are thus selected as a weighted measure of their "global" superiority, with more weight given to the worst class pairs and possibly no weight given to the best class pairs. Various methods of assigning the threshold values automatically as a function of the measurements in the pool were presented. A very important feature of this algorithm is the ability to vary the thresholds as a function of *a priori* knowledge concerning the features in the pool and the remainder of the system which will utilize the chosen measurements. In those cases where it is desirable to discriminate certain class pairs more than others, the thresholds are modifiable and can be individually varied.

From the results shown in Table 3, the threshold selection algorithm is uniformly superior to both generalized sequential selection and generalized sequential rejection. Only threshold selection was discussed, although threshold rejection would be straightforward to implement if conditions warrant.

A modification to the threshold selection procedure was made to include a measure of redundancy among the features selected. This modification employs a measure of the correlation of the feature selected (the Pearson correlation coefficient). The redundancy measure is used to reduce each feature's contribution to the overall utility of the subset. This modification may be made to any selection or rejection algorithm employing a distance measure. The threshold selection algorithm with this redundancy measure (Table 4) is superior to the threshold algorithm without the measure and is markedly super-

ior to generalized selection and rejection. The only disadvantage to including this measure is the computational and storage costs involved. However, in most cases, feature selection is only performed once and these costs may not be an important factor.

REFERENCES

1. S. E. Estes, "Measurement selection for linear discriminants used in pattern classification," IBM Research Report, RJ-331, April 1965.

2. E. M. Riseman and A. R. Hanson, "A Contextual Postprocessing System for Error Correction Using Binary n-Grams," IEEE Trans. on Computers, Vol. C-23, No. 5, May 1974, pp. 480-493.

3. A. R. Hanson, E. M. Riseman, and E. Fisher, "Context in Word Recognition," COINS Technical Report, University of Massachusetts, Amherst, September 1974, (to appear in Pattern Recognition).

4. R. C. Ahlgren, H. F. Ryan, and C. W. Swonger, "A Character Recognition Application of an Iterative Procedure for Feature Selection," IEEE Trans. on Computers, Vol. C-20, Sept. 1971, pp. 1067-1086.

5. G. D. Nelson and D. M. Levy, "A Dynamic Programming Approach to the Selection of Features," IEEE Trans. on Systems Science and Cybernetics, Vol. SSC-4, July 1968, pp. 145-151.

6. C. -L. Chang, "Dynamic Programming as Applied to Feature Subset Selection in Pattern Recognition Systems," IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-3, March 1973, pp. 166-171.

7. A. N. Mucciardi and E. E. Gose, "A Comparison of Seven Techniques for Choosing Subsets of Pattern Recognition Properties," IEEE Trans. on Computers, Vol. C-20, Sept. 1971, pp. 1023-1031.

8. W. S. Meisel, Computer-Oriented Approaches to Pattern Recognition. New York: Academic Press, 1972, Chapter 9.

9. K. Fukunaga, Introduction to Statistical Pattern Recognition, New York: Academic Press, 1972.

10. R. Bakis, N. M. Herbst, and G. Nagy, "An Experimental Study of Machine Recognition of Hand-printed Numerals," IEEE Trans. on Systems Science and Cybernetics, Vol. SSC-4, July 1968, pp. 119-132.

11. J. H. Munson, "Experiments in the Recognition of Hand-Printed Text," Proceedings of the 1968 Fall Joint Computer Conference, Vol. 33, Washington D.C.: Thompson, 1968, pp. 1125-1138.

12. F. Pratt, Secret and Urgent. Garden City, N.Y.: Blue Ribbon Books, 1942.

13. A. B. S. Hussain, G. T. Toussaint, and R. W. Donaldson, "Results Obtained Using a Simple Character Recognition System on Munson's Hand-printed Data," IEEE Trans. on Computers, Vol. C-21, Feb. 1972, pp. 201-205.

APPENDIX A: DESCRIPTION OF THE FEATURE POOL

The features chosen for inclusion in the pool reflect the types of features commonly found in the literature. Very little effort was made to tailor the feature pool to the problem domain described earlier. The distribution of these features is as follows:

52 randomly designed n-tuples
22 hand-picked n-tuples
45 high information n-tuples
8 topological measurements
25 windows

Clearly, the performance of the system is limited by the quality of the features contained in the pool; a more thoughtful choice of features will undoubtedly produce an increase in performance. However, this was not the subject of study in this paper.

The randomly designed and hand-picked n-tuple measurements are 5 x 5 matrices whose elements are zero, one, or "don't care." The features are "on" if a match is made at any point as the window is shifted over the character. The high-information n-tuples are masks which are either 2 x 3, 2 x 5, 3 x 2, 3 x 3, or 5 x 2. These are each placed in a particular location on the character and are "on" if the density (blackness) of that grid exceeds a given threshold. The topological features measure the contour of the character as viewed from the horizontal and vertical edges. Four of these features measure concavity and are binary features. The remaining four features measure the straightness of the outer contour. The line intersection features each count the number of times a line drawn through the character intersects the character. The windows are those of Hussain, *et al.*, [13]. The value of each measurement is the density of a 4 x 4 segment of the 24 x 24

character grid.

Table A.1 is a summary of the distribution of features selected from the pool for the sets of thirty features.

Type of Feature	random n-tuple	hand-picked n-tuple	hi-information mask	topological	line intersection	window
Sequential Selection						
m = 1	2	2	7	5	7	7
m = 50	0	3	6	7	11	3
Sequential Rejection						
m = 1	5	3	9	1	5	7
m = 50	1	0	8	5	9	7
Threshold Selection						
c = 1/2 (same as c = 2/3)	1	1	5	7	8	8
Threshold Selection with Redundancy Measure						
c = 1/2 (same as c = 2/3)	2	1	6	7	6	8

Table A.1. Distribution of features chosen for selected thirty feature subsets of P.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

University of Massachusetts, Amherst, MA 01002

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

None

3 REPORT TITLE

FEATURE SELECTION ALGORITHMS USING NON-REDUNDANT THRESHOLDED MEASURES

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5. AUTHOR(S) (First name, middle initial, last name)

Edward G. Fisher
Allen R. Hanson
Edward M. Riseman

6. REPORT DATE

December 1974

7a. TOTAL NO. OF PAGES

27

7b. NO. OF REFS

13

8a. CONTRACT OR GRANT NO.

N00014- 67-A-0230-0007

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

COINS Technical Report 74C-9

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES

None

12. SPONSORING MILITARY ACTIVITY

Office of Naval Research, Code 437
Washington, D.C.

13. ABSTRACT

A new feature selection method, the threshold selection algorithm, is presented and compared with sequential selection and rejection algorithms. This algorithm assumes a measure of feature discrimination exists and provides a set of threshold parameters, associated with class pairs, which are dynamically variable. These thresholds provide a local as well as a global perspective to the problem of selection of feature subsets from a pool. Each threshold parameter provides an upper limit of class separation to be attained during feature selection; once this limit is reached, that class pair no longer affects the selection of features. Thus, the procedure maintains a global perspective by considering equally all class pairs which have not achieved their thresholds and in addition, particularly during the latter steps, focuses on the fewer local cases which have not been discriminated sufficiently.

The basis of comparison of the algorithms is a pattern recognition system operating on hand-printed alphabetic characters. The threshold selection algorithm provides improvement (in terms of system error rate) over sequential selection and rejection. Finally, a modified threshold selection algorithm with a redundancy measure is described which exhibits a considerable improvement in performance.