A PROBABILISTIC ASSOCIATION MEASURE FOR PATTERN RECOGNITION

by

Xiaobo Li

A DISSERTATION

Submitted to

Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science

1984

ABSTRACT

A PROBABILISTIC ASSOCIATION MEASURE
FOR PATTERN RECOGNITION

by

Xiaobo Li

This thesis investigates the properties of a probabilistic association measure in four areas of pattern recognition and image processing. It is essential to measure the association between patterns, between features and between partitions in both pattern recognition and exploratory pattern analysis. Direct interpretation and known distribution are desirable properties for an association measure. This thesis proposes a statistic with these characterisitcs under a permutation null hypothesis which has been used to advantage in the literature. The four problems attacked in this thesis are described below.

A preliminary feature analysis studies the unusualness of feature-category relations before feature extraction, which is just as important as cluster tendency study before clustering. This is formally defined in this thesis for the first time in a statistical hypothesis testing framework. This thesis shows that the permutation statistic is preferable to the commonly used correlation coefficient with binary features. Both statistics have comparable power but

the threshold of the permutation statistic has a more direct interpretation than that of the correlation coefficient. The use of this statistic is demonstrated on questionnaire data.

The second and third problems attacked with the permutation statistic are the measurement of the adequacy of binary partitions in validating clustering results and designing tree classifiers. The relations among three well-known measures of cluster validity are derived and the permutation statisti is shown to be different but highly correlated to these measures. Computational advantages make it preferable to the other statistics. In tree classifier design, the permutation statistic can be used as a criterion for choosing the feature and threshold at each node in the tree. The threshold can be set more easily than with the mutual information criterion. Several examples on artificial data and real data sets demonstrate that the permutation statistic is a reasonable criterion for node definition and leads to successful tree classifiers.

The fourth problem investigated is the general problem of image template matching posed as a test of hypothesis. The alternative hypothesis is that a distorted version of the object is in the image. The null hypothesis is that the object is not in the image. We derive an approximate likelihood ratio statistic for testing these hypotheses and

compare it to three other statistics. The optimal statistic is most powerful but computationally intensive. A simplified Neyman Pearson statistic is shown to be more powerful than other suboptimal measures, yet to remain sensitive to the true object location in the image. The permutation statistic acts about the same as other sub-optimal measures, such as absolute difference and correlation coefficent.

The last chapter of the thesis reviews algorithms for computing cumulative hypergeometric distribution functions, which is essential to the proposed statistic. A pipeline architecture design implementing a recursive computation formula is proposed. This design is more efficient than other exact computation methods by several degrees of magnitude. It is faster than the best approximation algorithm for many reasonable cases.

To my father and my mother

ACKNOWLEDGEMENTS

I thank my thesis advisor, Professor Richard C. Dubes, for his help, guidance and encouragement throughout the preparation of this thesis. He has introduced me to careful scientific research and precise scientific writing and he has contributed greatly to my interest in Pattern Recognition and my research abilities. His continuing direction was essential to this study.

Thanks are also due to Professor Anil K. Jain and Professor George Stockman for their helpful discussions and guidance in the research group and my degree committee; to Professor Raoul LePage for his important suggestions on the theoretical work of this thesis; to Professor Carl Page for his advice and excellent teaching of many of my courses. Special thanks to Professor Lionel M. Ni for his brilliant help on the hardware implementation part of this thesis.

Acknowledgement are due in addition to past and present graduate students of the MSU PR/IP Laboratory for their assistance: Dr. George Cross, Dr. Weichung Lin, Dr. Steve Smith, Dr. James Coggins, Dr. Gautam Biswas, Dr. Ardeshir Goshtasby, Almost-Dr. Rick Hoffman, Phil Nagan,

and Juan Esteva. System manager Phil Nagan's maintenance of the laboratory greatly facilitated this thesis work. A special acknowledgement to my office mate Rick Hoffman for his theoretical discussions and creating useful system software which played a large role in preparation of this thesis.

I must thank my wife Zhemin for her sharing my burden and difficulties. Her understanding and support, the sacrifice of my son and my family made this study possible.

The financial support of NSF grants ECS-800716 and ECS-83002004 is also gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Summary

This thesis proposes a probabilistic association measure for binary vectors and examines its application in pattern recognition. It is essential to measure association between patterns, between features, between partitions of patterns or between subimages. There exist many association measures in the pattern recognition literature. Only Goodall's similarity index is directly based on probability. This Chapter reviews previous work on probabilistic measures and the permutation hypothesis and proposes a permutation statistic S, which is directly based on permutation unusualness and has known distribution under randomness.

In Chapter 2, we apply S to the preliminary analysis of binary features for the first time and state its utility. We define several null hypotheses expressing the relation

between features and category. Rejection of these hypotheses motivates feature extraction. We compare S to Pearson correlation for testing these hypotheses. Since S has known distribution under all null hypotheses, the selection of thresholds for the tests is direct. The selection of the threshold for Pearson correlation demands Monte Carlo simulation. Both statistics have similar power for detecting Bahadur type alternative hypotheses and both act similarly in a questionnaire data analysis example. This suggests S be used in the preliminary analysis of binary features.

Chapter 3 applies the permutation statistic $S'=|S-0.5|$ as an adequacy measure for binary partitions in cluster validity and as a criterion for tree classifier design. Three statistics, Rand, Hubert's G_{amma} and Fowlkes' B, are compared to S' in the permutation environment. These three statistics are linear functions of each other and thus have the same power against any alternative hypotheses in testing the random permutation null hypothesis. The statistic S' is highly correlated to those statistics when the numbers of 1's in the vectors are about half the vector length. The other three statistics have asymptotic normal distributions under the null hypothesis while S' has a known distribution. That makes S' easier to use for assessing global fit of a

binary partition to category than the other three statistics. The statistic S' is used for the first time in designing a tree classifier, and is compared to a mutual information criterion. Both criteria give similar trees but the threshold for S' is far easier to determine. Several artificial data sets and a real data set are used in the numerical examples. The results demonstrate that S' is a reasonable measure in tree classifier design.

Chapter 4 investigates the image template matching problem. We study a null hypothesis stated in the literature and propose a new alternative hypothesis. Under this model, we propose a statistic R which is derived directly from the likelihood ratio statistic to measure similarity between template and subimages. With an extra independence assumption, this statistic is theoretically optimal, but its computation is very time consuming. We suggest a simplified version of the statistic, G, as a similarity measure and compare the powers of these two statistics by a Monte Carlo means to the commonly used similarity measure, the absolute difference D. The statistic R is most powerful. The G statistic is more powerful than other sub-optimal measures and has the same complexity as others. More importantly, G is more sensitive to the true location of the object in the image than any

others. The correlation coefficient C and the permutation statistic S defined in Chapter 1 are also compared to the likelihood statistics and they perform about the same, worse than the derived measures. All these statistics are applied to several Landsat data sets and demonstrate that R is not as sensitive as statistic G. In summary, G strikes the best balance between power ans sensitivity.

Since the computation of S is directly based on the hypergeometric distribution, Chapter 5 reviews some commonly used algorithms for computing the hypergeometric c.d.f. Their computation times are compared and a pipeline architecture design is proposed to implement the recursive computation algorithm. The pipeline design is much more efficient than other exact computation methods. It is even faster than the Peizer approximation using single CPU when the vector size is small or a, the number of (1,1) pairs, is small ($a < 932$). This pipeline design also solved a more general problem. In implementing a product using pipeline functional units with x segments, we give examples of some individual designs for different y values and one general design for $y=2,3,\ldots,15$, which are typical pipeline length for arithmetic operators. A computation time and circuit complexity analysis are given.

From the above applications, we conclude that the statistic S is a good similarity measure between features in preliminary analysis for binary features; S is a good similarity measure between feature and category in tree classifier design; and S is also a reasonable adequacy measure for binary partitions. Since S has a known distribution under null hypotheses, the threshold and significance level are easy to determine. In image template matching, S acts as well as other sub-optimal similarity measures between sub-images. Finally, a hardware architecture design is more efficient than other methods for computing hypergeometric c.d.f.'s, which is essential to the computation of S. The application of S and other proposed measures to the above pattern recognition areas and the methodology for using S are the main contributions of this thesis.

1.2 Background

Pattern Recognition is concerned with the description and classification of objects, which are represented by measurements taken from realizations of physical processes. Pattern recognition includes three distinct steps -

Representation, Abstraction, and Generalization. The input data for pattern recognition is usually represented as a pattern matrix, whose rows are patterns. Each pattern consists of a series of measurements, called features, and describes an object. In classical pattern recognition problems, the category, or pattern class, of each training pattern is known. A subset of the features or some function of this subset is extracted for the representation of the patterns. Based on this subset and category information, a pattern classifier is designed. Feature extraction and classifier design are the Abstraction phase of Pattern Recognition. The Generalization phase of Pattern Recognition involves evaluation of algorithms and classifiers.

There are two mathematical approaches to solving pattern recognition problems: the decision-theoretic (or discriminant) approach and the syntactic (or structural) approach. In the decision-theoretic approach, each pattern is considered as a vector in a feature space and the recognition is made by partitioning the feature space. In the syntactic approach, each pattern is considered as a composition of components, called sub-patterns or primitives, and the recognition is made by parsing the pattern according to some syntax rules.

Cluster analysis and image processing are closely related to pattern recognition. Cluster analysis explores the data structure with and without category information, which can be considered as the Representation phase of Pattern Recognition. Image processing deals with two dimensional pictorial patterns. Many basic techniques used for pattern recognition and image processing are very similar in nature.

This thesis proposes an association measure for binary vector pairs based on permutation statistics. Its application in various areas of pattern recognition is compared to other measures of association and both hardware and software computational methods are developed. The particular areas of application studied are preliminary analysis of binary features, validity of binary partitions, design of tree classifiers, and image template matching. Section 1.3 reviews previous work in probabilistic proximity measures and permutation models. Section 1.4 defines the proposed statistic.

1.3 Probabilistic Proximity Measures and Permutation Models

A proximity (or association) measure, either a similarity or a dissimilarity, is a symmetric mapping from VxV to $[0,\infty)$, where V is a space of vectors. A similarity measure increases as two vectors become more alike, while a dissimilarity measure increases as two vectors become more unalike. For example, Euclidean distance is a dissimilarity measure and correlation is a similarity measure. The history of Statistics is replete with association measures [84]. Hundreds of proximity measures have been proposed by researchers in the biological and engineering literature. Comprehensive reviews [1,25,26,27,28,77] and comparative experiments have been reported [21,30,41,71].

This thesis is mainly interested in measures using cumulative probability, not information theory, although some similarity measures based on information theory are also considered to be probabilistic measures [77]. Goodall's similarity index is the best example of this type [22,23,24]. We seek a statistic which can be easily interpreted and is easy to compute.

A population is needed to establish the significance of a probabilistic index of association. The null hypothesis is that all members of the population are equally likely. A permutation population is formed from all permutations of the components of the vectors. The vectors could be patterns, features, subimages or partitions.

Permutations have long been used as a randomization procedure to generate populations [44]. Statistics useful in pattern analysis, such as the Mantel statistic [19,33,35,37,53,54] and the B statistic [17] use permutations to generate null hypotheses. These two indices have known asymptotic distributions under this null hypothesis. Formulas for exact means and variance are available but require a significant amount of computation.

The permutation null hypothesis for a vector pair used for much of our work is defined below.

H_0: all permutations of one vector in the pair are equally likely.

If this vector pair represents partitions of a pattern set resulting from clustering, then the testing of this hypothesis is a cluster validity analysis, which validates the global fit of a partition. This null hypothesis is reasonable for external criteria, as when prior structure is being compared to the result of a cluster analysis [33]. However, it is irrelevant as an internal criterion [18,42,62] even though it is the only hypothesis for which analytical results are available. Specific null hypotheses, alternative hypotheses, and appropriate tests are stated in each area of investigation.

For some statistics, Monte Carlo simulation is needed to obtain the critical levels, thresholds or power estimates. Usually, we set the size of the Monte Carlo simulation to 100 [11]. In some permutation situations, we set the size of Monte Carlo simulation to 1000 [33].

1.4 Proposed Probabilistic Indices for Binary Vector Pairs

If a statistic measuring association between data items has known distribution under some H_0, its tail probability (the probability that this statistic is less than or equal to a threshold) is called a p-value, and is a probabilistic index of association. This section proposes the p-value of the number of matches between two binary vectors as such an index. This index has known distribution under H_0.

Consider a pair of binary vectors, V_1 and V_2, of size L, whose entries are either 0 or 1. The number of 1's in V_i is n_i. The set of all possible permutations of V_2 forms a sample space. Table 1.1 defines the observables for these two vectors. For example, a' is the number of positions in which both vectors are 0 and n_1 is the number of 1's in V_1.

Table 1.1 Frequencies of Combinations of Two Binary Vectors

V_1 \ V_2	0	1	
0	a'	b'	$L-n_1$
1	c'	d'	n_1
	$L-n_2$	n_2	L

The random variable D' denotes the number of positions in which the two vectors are both 1. For example, If V_1=(100110) and V_2=(110010) then D'=d'=2. The following statistic measures the degree of association between two vectors.

$$S(1,2) = Pr(D' < d') + Pr(D' = d')\ U$$

where U is an independent random variable with uniform distribution over [0,1]. Probabilities are computed under H_0.

Under H_0, $S(V_1,V_2)$ is distributed uniformly over [0,1]. It has a clear interpretation as permutation unusualness, i.e., the probability that a vector pair has at most that many (1-1) matches. The explicit formula for the statistic

is as follows.

$$S(1,2) = \sum_{i=0}^{d'} \frac{\binom{n_1}{i}\binom{L-n_1}{n_2-i}}{\binom{L}{n_2}} - \frac{\binom{n_1}{d'}\binom{L-n_1}{n_2-d'}}{\binom{L}{n_2}} U$$

This statistic is used as a similarity measure between features in preliminary feature analysis (Chapter 2), as a similarity measure between partitions in measuring partitional adequacy (Chapter 3), and as a similarity measure between subimages in image template matching (Chapter 4). The null hypotheses are rejected for extreme values of this statistic.

CHAPTER 2

PRELIMINARY ANALYSIS OF DICHOTOMOUS FEATURES

Feature selection in pattern recognition chooses a subset of features which best reflects a-priori category information in some manner [78] and which are "independent" in some sense. Feature selection algorithms select or extract features one at a time and stop when a predefined threshold has been exceeded. For example, the percent variance retained in the eigenvector method of feature extraction [78] and the recognition rate in the sequential forward method of feature selection [86] are thresholds which the user must specify beforehand. Few theoretical guidelines exist for selecting these thresholds, especially when the relationships among the variables are completely arbitrary. This chapter proposes a methodology for examining a data set before feature extraction to alleviate the danger of interpreting the results of feature extraction inappropriately. For example, we ask whether or not binary (dichotomous) features significantly relate to each other,

and whether or not some features significantly relate to category. If no significant relations exist, it would appear useless or misleading to proceed with feature extraction and discriminant analysis. The motivation for this preliminary analysis is much the same as for clustering tendency [13] where the pattern set is examined for randomness.

In this chapter, we pose several questions about dichotomous features in a statistical hypothesis testing framework. We define hypotheses and examine the application of the probabilistic similarity measure S, defined in Chapter 1, for testing those hypotheses, in comparison with statistics based on Pearson correlation.

An important application of the proposed methodology is to questionnaire data, which come as an NxK pattern matrix containing only 0's and 1's. Rows denote patterns and columns denote features. Some of the features may be reserved as category information.

In Sec. 2.1, we examine the entire pattern matrix in terms of statistical hypotheses, and define two statistics for establishing the existence of relationships between columns. One is based on the correlation coefficient and

the other is based on the similarity S. We discuss their distributions and tests of hypotheses. Section 2.2 asks if any feature is related to category. We briefly discuss the distributions of two statistics. Sec. 2.3 describes approximations to the distributions of the statistics defined in Sec. 2.1 and 2.2, based on the similarity S. Section 2.4 reports a power comparison of these statistics. The methodology is demonstrated on a data set derived from medical questionnaires in Sec. 2.5. Finally, Sec. 2.6 summarizes the results in this chapter.

The main contribution of this chapter is the methodology for the preliminary analysis of binary features, which is summarized in Sec. 2.6.

2.1 Are Any Two Features Unusually Similar to One Another?

In this section, we consider the first question in the preliminary analysis of dichotomous features stated below. Two null hypotheses and their tests will be defined. The distributions of test statistics will be discussed in some detail.

Consider an NxK binary pattern matrix in which columns can be considered as categorical or measured features. This first question searches for significant similarity between any pair of features, including those used later as categories. Two reasonable hypotheses for describing a state of "no relationship" are defined below. In all cases, we regard the rows of the pattern matrix as samples of independent random variables.

H_{01}: The patterns are samples of K independent Bernoulli random variables with parameters $\{p_i\}$.

H_{02}: The pattern matrix is chosen randomly from population P_2.

Population P_2 is formed by independently permuting each column in the pattern matrix. There are $(N!)^K$ matrices in P_2. If N_i denotes the number of 1's in column i, then the number of distinct matrices in P_2, each occuring the same number of times, is

$$\prod_{i=1}^{K} \frac{N!}{[N_i!(N-N_i)!]} .$$

If a matrix can be realized under both hypotheses, its probability is lower under H_{01} than under H_{02}. An ideal statistic for testing H_{01} or H_{02} satisfies the following criteria:

1. It has reasonable power against certain alternatives.
2. Its value is easy to compute.
3. Its distribution is known analytically, or at least can be approximated analytically.
4. The analytical form is simple.
5. Its distribution is independent of the parameters in the problem.

Several measures of correspondence between dichotomous variables have been suggested in the literature [1,30,77]. In this chapter, we examine the application of the similarity S to test H_{01} and H_{02}. We define statistic $\underline{S}$ directly based on S below. We also define statistic $\underline{C}$ based on the commonly used Pearson correlation coefficient for purpose of comparison.

$$\underline{C} = \max\{c(i,j)\}$$
$$\underline{S} = \max\{S(i,j)\}$$

Here c(i,j) is the sample Pearson correlation between column

i and j, and S(i,j) is defined in Chapter 1.

Let N_i be the number of 1's in column i and let N_{ij} be the number of rows in which columns i and j are both 1.

$$c(i,j) = (NN_{ij}-N_iN_j)/\ [N_iN_j(N-N_i)(N-N_j)]^{1/2}$$

In our case, S(i,j) takes the following form:

$$S(i,j) = [\sum_{---}^{---} h(N,N_i,N_j,y)]-h(N,N_i,N_j,N_{ij})U$$

where the sum is for y from $\max(0,N_i+N_j-N)$ to N_{ij}, U is a (continuous) uniform random variable on the unit interval, independent of all other random variables, and h is the density function for a hypergeometric distribution, defined below.

$$h(N,K,L,y) = \frac{\binom{K}{y}\binom{N-K}{L-y}}{\binom{N}{L}}$$

Note that since h(N,K,L,y) = h(N,L,K,y), including 0-0 matches as well as 1-1 matches does not alter the nature of the statistic.

Large positive values of c(i,j) and S(i,j) indicate a positive relation between features i and j. With the questionnaire type applications in mind, we only consider

positive relations because the coding in questionnaire data is usually fixed and (1,1) matches have stronger meaning than (0,0) matches.

Under H_{02}, N_{ij} has a hypergeometric distribution and c(i,j) is a linear function of N_{ij}. The distribution of c(i,j) has a rather complicated form so the analytical forms for the distributions of $\underline{C}$ under either H_{01} or H_{02} are also complicated. The threshold of this distribution is usually estimated by Monte Carlo means. The distribution of S(i,j) under H_{02} is clearly uniform. The fact that S(i,j) is uniformly distributed over [0,1] under H_{01} follows from the following equality.

Write S(1,2) as an explicit function of random variables N_1, N_2, N_{12} and U. The sum is over all allowable values of (z_1, z_2).

$$\begin{aligned}
&\Pr(S(N_1,N_2,N_{12},U) \le y \mid H_{01}) \\
&= \sum_{z_1}\sum_{z_2} \Pr(S(z_1,z_2,N_{12},U) \le y \mid H_{02}) \Pr(N_1=z_1, N_2=z_2 \mid H_{01}) \\
&= y \sum_{z_1}\sum_{z_2} \Pr(N_1=z_1, N_2=z_2 \mid H_{01}) \\
&= y
\end{aligned}$$

Based on the fact that S(i,j) has a uniform distribution under H_{01} and H_{02}, Sec. 2.3 derives a simple approximation to the distribution of $\underline{S}$ under H_{02} for M>2, which eliminates Monte Carlo work when choosing thresholds for tests of hypothesis.

A test of H_{01} or H_{02} has the following form, where T is either $\underline{C}$ or $\underline{S}$.

$$\text{Reject } H_0 \text{ if } T>t$$

where threshold t is computed as

$$P(T>t|H_0) = a$$

and a is a specified level, such as 0.05. Alternatively, we compute the critical level a^*

$$P(T>t^*|H_0) = a^*$$

where t* is the observed value of the statistic and reject the null hypothesis if a^* is small, say 0.1 or less.

The distribution of $\underline{C}$ does not have a simple analytical form, and depends on the parameters $\{p_i\}$ under H_{01} and on the original matrix under H_{02}. The distribution of S(i,j) has a simple analytical form, regardless of parameters, under both H_{01} and H_{02}. An approximation for the distribution of $\underline{S}$ will be developed. Therefore, according

to Criteria 2 through 5, S is prefered over C . The comparison according to Criterion 1 is discussed in Sec. 2.4.

2.2 Is Any Feature Unusually Closely Related to Category?

Here we designate one feature, say feature M, as categorical and ask whether any one of the remaining features is unusually similar to feature M. Two null hypotheses of interest are defined below.

H_{03}: All N! permutations of feature M are equally likely.

H_{04}: All matrices in population P_4 are equally likely.

Population P_4 is created in the same way as population P_2 except that feature M is not permuted. Two test statistics based on our indices of proximity are:

$$C_M = \max\{c(i,M), i \neq M\}$$

$$S_M = \max\{S(i,M), i \neq M\}$$

The forms of tests of these hypotheses are as in Sec. 2.1. A test of H_{03} asks whether it is likely that the observed statistic could have been produced were the

category labels inserted at random. Accepting H_{03} implies that no feature is unusually "close" to the categorical feature. The distribution of C_M under H_{03} must be obtained by Monte Carlo means, but a procedure discussed in Sec. 2.3.3 can approximate the distribution of S_M under H_{03}.

All columns are permuted independently so the distributions of C_M and S_M are known under H_{04}.

$$P(S_M>t \mid H_{04}) = 1 - t^{(K-1)}$$

$$P(C_M>t \mid H_{04}) = 1 - \prod_{i\neq M} P[\, c(i,M)\leq t \mid H_{04}]$$

where

$$P[c(i,M)=t_i \mid H_{04}] = h(N,N_i,N_M,t_i)$$

and $t_i = [N_iN_j+t[N_iN_j(N-N_i)(N-N_j)]^{1/2})]/N$

Under H_{03}, the distribution of S_M can be approximated analytically, while the distribution of C_M requires Monte Carlo simulation. Under H_{04}, both statistics have known distributions, but that of S_M is uniform. This fact suggests that S_M is better than C_M under Criteria 3 through 5.

2.3 Approximating the Distributions of $\underline{S}$ and S_M

We first discuss the distribution of $\underline{S}$ when K=2. Then we view $\underline{S}$ as the maximum of dependent uniform random variables and try to approximate its distribution. The distribution of S_M is then considered.

2.3.1 Null Distributions of $\underline{S}$ for K=2

Assume K=2 so that $\underline{S}$ is simply the similarity measure between two binary vectors; $\underline{S}$ is uniform under H_{01} regardless of N, p_1 and p_2. Table 2.1 demonstrates empirical means and variances for $\underline{S}$ and $\underline{C}$. Each row in Table 2.1 corresponds to a parameter set (N,M,p_1,p_2). One hundred Monte Carlo samples were used for each row.

Table 2.1. Means and Variances of $\underline{C}$ and $\underline{S}$ under H_{01} for K=2.

N	p_1	p_2	E($\underline{C}$)	V($\underline{C}$)	E($\underline{S}$)	V($\underline{S}$)
20	.10	.10	-.02808	.03129	.51732	.08919
20	.10	.50	-.01565	.04664	.53185	.08632
20	.10	.90	.01468	.04015	.48699	.08465
20	.50	.50	.02982	.05273	.48520	.07934
20	.50	.90	.00577	.03625	.54219	.06516
20	.90	.90	.01331	.03786	.54297	.08038
200	.10	.10	.00578	.04061	.45545	.07565
200	.10	.50	.05577	.08443	.47903	.07233
200	.10	.90	.00137	.04455	.49546	.09711
200	.50	.50	-.07583	.09182	.47418	.08335
200	.50	.90	-.05822	.23322	.50266	.08883
200	.90	.90	.00896	.04410	.52179	.07967

The theoretical mean and variance of $\underline{S}$ under H_{01} are 0.5 and 0.0833. The empirical means and variances of $\underline{S}$ in Table 2.1 are stable and close to theoretical values, independent of N, K, p_1 and p_2. The distribution of $\underline{C}$ varies with these parameters. This fact favors $\underline{S}$ over $\underline{C}$ with respect to Criterion 5.

2.3.2 Approximating the Null Distribution of $\underline{S}$ when K>2

Statistic $\underline{S}$ is a maximum of k=K(K-1)/2 dependent U(0,1) random variables under H_{01}. A one-sided test of H_{01} is

Reject H_{01} if $\underline{S} > t_+$.

The dependences among {S(i,j)} require that t_+ be estimated from an approximate null distribution for $\underline{S}$. Since the distribution of the maximum of k independent random variables is well known, we consider the possiblility of approximating t_+ from this distribution. Let T' have this distribution, whose c.d.f. is

$$Pr(T' \leq t' \mid H_{01}) = t'^{k} \quad \text{if } 0<t'<1.$$

The threshold t' from this distribution at size a is $t'=(1-a)^{1/k}$. Table 2.2 gives the thresholds t_+ for $\underline{S}$ generated by 1000 Monte Carlo runs per entry ($p_i=0.5$ for all i), and the thresholds t' computed analytically for a few values of K. The right-most column in Table 2.2 gives the position of t' among 1000 Monte Carlo values, which is an estimate of the size for the test using t'.

Table 2.2 0.05 Thresholds for $\underline{S}$ and T'

N	K	t_+	t'	size for t'
50	3	.98045	.98305	.045
	6	.99650	.99659	.049
100	6	.99635	.99659	.048
	9	.99845	.99858	.045
200	9	.99827	.99858	.042
	13	.99924	.99934	.044

The values of t' are close to those of t_+. Thus using t' as an approximate threshold will give similar size for the test.

In order to more formally compare the distributions of $\underline{S}$ and T', Consider the 12 cases listed in Table 2.3. One hundred matrices were generated for each case and the distributions of $\underline{S}$ and T' were compared. The results of the Kolmogorov-Smirnov test of equality between distributions [7] are listed below. In all 12 cases, the hypothesis that T' is the same as $\underline{S}$ is accepted at level 0.05. This suggests that the threshold on T' is indeed a good approximation to the threshold on $\underline{S}$ for those cases.

Table 2.3 Comparing Distributions of $\underline{S}$ and T'

case	N	K	p_1	p_2	KS stt
1	20	5	.1	.1	.059
2	20	5	.1	.5	.024
3	20	5	.1	.9	.039
4	20	5	.5	.5	.043
5	20	5	.5	.9	.029
6	20	5	.9	.9	.033
7	200	5	.1	.1	.061
8	200	5	.1	.5	.043
9	200	5	.1	.9	.041
10	200	5	.5	.5	.103
11	200	5	.5	.9	.051
12	200	5	.9	.9	.061

Statistic S_M is a maximum of (K-1) dependent U(0,1) random variables under H_{03}. When K>2, our results suggest that the distribution of T' approximates the distribution of S_M. This point is also demonstrated in the application described in Sec. 2.5.

2.4 Power Comparison of $\underline{S}$ and $\underline{C}$

The purpose of using $\underline{C}$ and $\underline{S}$ is to detect relations among features. We now examine the powers of these statistics under a class of alternatives where the relations among features are governed by a Bahadur model [4], in which the feature relations are specified in terms of the correlation coefficient. In this section, we consider the following alternative hypothesis.

H_{11}: The given matrix is generated with Bahadur parameter set ($p_1,\ldots,p_K$ and w) with w>0; p_i is the Bernoulli parameter for column i.

Power studies are conducted for the case when p_i under H_{01} is the same as the p_i under H_{11} for any i, so that the hypotheses differ only in relation among features.

The test of H_{01} relative to H_{11} based on $\underline{S}$ is:
Reject H_{01} if $\underline{S} > t_s$.

The test relative to H_{11} based on $\underline{C}$ is:
Reject H_{01} if $\underline{C} > t_c$.

The general procedure for estimating power of a test based on $\underline{C}$ is as follows [16]. One hundred matrices are generated under H_{01} with given p's and the sixth largest value is used as the 0.05 level t_c. Then, 100 matrices are generated under H_{11} with given $w \neq 0$. Our study involves the four factors listed below with levels for each factor indicated.

$(p_1,p_2) = (.2,.2),(.2,.5),(.2,.8),(.5,.5),(.5,.8),(.8,.8)$;
$p_i = 0.5$ for $3 \leq i \leq K$;
$w = 0.1, 0.2, 0.4$;
$N = 20, 60, 200$;
$K = 2, 5$.

All combinations of levels are used except the impossible case $(p_1, p_2, w) = (.2,.8,.4)$. The power of the test based on $\underline{C}$ is estimated by the number of matrices generated under H_{11} with $\underline{C}$ value greater than t_c. The power of the test based on $\underline{S}$ is estimated similarily, except that the

threshold t_s is approximated analytically (Sec. 2.3.2).

For each combination of parameters, we have two integers, the estimated power (P"($\underline{C}$)) of the correlation statistic and the estimated power (P"($\underline{S}$)) of S. Let P($\underline{C}$) and P($\underline{S}$) denote the true powers. Then P"($\underline{C}$) has a binomial distribution B(100, P($\underline{C}$)) and P"($\underline{S}$) has a B(100, P($\underline{S}$)) distribution. We report P"($\underline{C}$) and P"($\underline{S}$) in Table 2.4 for K=2 and in Table 2.5 for K=5. The symbol "+" at the upper right corner indicates that P($\underline{S}$) > P($\underline{C}$) at confidence level 0.95. The symbol "-" means P($\underline{S}$) < P($\underline{C}$) at confidence level 0.95.

Table 2.4 Power Estimates when K=2

N, w	(p_1,p_2)=(.2,.2)	(.2,.5)	(.2,.8)	(.5,.5)	(.5,.8)	(.8,.8)
20, 0.1	12 11	18 12	5 + 14	8 7	11 13	10 12
20, 0.2	26 21	30 21	10 16	19 19	21 26	17 25
20, 0.4	54 - 42	65 - 46		56 55	57 51	41 48
60, 0.1	17 19	33 - 17	14 17	15 15	22 22	14 19
60, 0.2	38 39	60 - 49	38 40	47 46	46 49	36 43
60, 0.4	85 86	97 93		94 94	93 95	86 89
200, 0.1	54 - 36	55 - 36	41 41	27 36	30 + 44	34 41
200, 0.2	96 91	95 91	98 98	89 90	76 + 84	74 81
200, 0.4	100 100	100 100		100 100	100 100	100 100

P"(C)
P"(S)

Table 2.5 Power Estimates when K=5

N, w \ (p_1, p_2)	(.2,.2)	(.2,.5)	(.2,.8)	(.5,.5)	(.5,.8)	(.8,.8)
20, 0.1	9 / 13	7 / 8	5 + / 11	4 / 7	4 / 9	8 / 7
20, 0.2	13 / 15	10 / 13	3 + / 9	9 / 11	4 + / 11	8 / 7
20, 0.4	36 - / 24	21 / 23		22 / 21	13 / 19	24 / 20
60, 0.1	6 / 7	5 / 6	5 + / 12	8 / 9	4 + / 13	12 + / 22
60, 0.2	16 / 21	13 / 18	5 + / 15	15 / 16	11 + / 29	28 / 31
60, 0.4	59 / 58	73 / 76		70 / 74	62 + / 81	64 / 67
200, 0.1	18 / 12	16 / 18	15 / 15	22 / 15	13 / 12	17 / 15
200, 0.2	69 - / 53	55 / 59	87 / 81	68 / 66	61 / 54	60 - / 45
200, 0.4	100 / 100	100 / 100		100 / 100	100 / 100	100 / 99

P"(C)
P"(S)

Table 2.4 and 2.5 are summarized into Table 2.6. Each box in Table 2.6 corresponds to a combination of K, N and w. In each box, the number at the upper left corner is the number of p' combinations (columns in Table 2.4 or 2.5) for

which P(C) > P(S) (minus signs) at level 0.95. The number at the lower right corner is the number of cases for which P(C) < P(S) (plus signs).

Table 2.6 Result of Power Comparison

K	2			5		
N	20	60	200	20	60	200
w						
0.1	0 / 1	1 / 0	2 / 1	0 / 1	0 / 3	0 / 0
0.2	0 / 0	1 / 0	0 / 1	0 / 2	0 / 2	2 / 0
0.4	2 / 0	0 / 0	0 / 0	1 / 0	0 / 1	0 / 0

Table 2.6 indicates that, among the 13 combinations of (K,N,w), for some (p_1,p_2) values P(C) and P(S) are significantly different. Seven combinations of (K,N,w) contain more cases for which P(S) > P(C) , while in six combinations the reverse is true. We conclude that S is as powerful as C under the condition of this experiment, even though the structure of the data under H_{11} was specified in terms of correlation. Although the powers of S_M and C_M for testing H_{03} and H_{04} were not compared experimentally, we expect them to be related in the same way.

The analytical approximations to the distributions for $\underline{S}$ and S_M are independent of parameters. In addition, they have the same power as $\underline{C}$ and C_M against certain alternatives. We conclude that $\underline{S}$ and S_M are prefered over $\underline{C}$ and C_M.

2.5 An Application of Preliminary Feature Analysis

As a detailed application of our methodology, we studied a set of questionnaire responses completed over the past several years by female patients of a medical doctor regarding self breast examination. We randomly chose 145 questionnaires for this study, no two from the same individual. We then reduced the data to 26 features. Table 2.7 defines the meaning of a "1" value for the first 13 features. A "1" value for features 14 through 26 denote the occurrence of cancer in a relative, such as a father, mother, or aunt.

Table 2.7 Definitions of First 13 Features

Feature	Meaning of "1" Value
1	Menstrual period has stopped
2	Operative menapause
3	Pregnant at least once
4	At least one miscarriage
5	Have used female hormones
6	Currently use female hormones
7	Positive result on pap smear
8	Perform semi-annual self breast exam
9	Perform monthly self breast exam
10	Have had mammograms
11	Use contraceptive techniques
12	Have had pelvic surgery
13	Have had breast surgery

The ultimate goal was to discover any factors that explained why self breast examination was performed by some women but not by others. Features 8 and 9 in Table 2.7 will be taken as category variables. This is a typical feature extraction problem in which a subset of features that predict category well is to be found. Some clustering algorithms were used to cluster features [40] and the resulting dendrograms suggested random clusters. A preliminary feature analysis should indicate whether any non-random relation exists among features.

Our experiments proceeded as follows.

1. Test H_{01} using $\underline{C}$ and $\underline{S}$.
2. Test H_{02} using $\underline{S}$.
3. Test H_{03} and H_{04} using C_M and S_M for two categorical

features.

4. Verify the results by a Mantel test [53] on original pattern matrix.
5. Perform feature extraction for any non-random categories.
6. Verify the results by Mantel tests on selected features.
7. Perform feature extraction for any categorical feature for which either H_{03} or H_{04} is accepted.
8. Verify the results by a Mantel test on selected features.

The details are explained in the following sections.

2.5.1 Tests for Significant Relations between Features

First, we test H_{01} and H_{02} using statistics $\underline{C}$ and $\underline{S}$. Critical levels for $\underline{C}$ were estimated from 1000 Monte Carlo trials. Random matrices were generated under H_{01} using N_i/N from the original pattern matrix to estimate p_i and $\underline{C}$ was computed for each matrix. The critical level for $\underline{S}$ was approximated by t' as explained in Sec. 2.3 and also estimated from 1000 Monte Carlo trials. Table 2.8 summarizes the results. The approximate critical level for $\underline{S}$ using t' is the same as the Monte Carlo result in this case. We reject H_{01} and H_{02} and conclude that the

questionnaire data merit further study.

Table 2.8 Testing H_{01} and H_{02}

Hypothesis	Stat, Value	Cr. Level a^*	Method
H_{01}	$\underline{C}$, 0.4528	0.001	1000 Monte Carlo Trials
H_{02}	$\underline{C}$, 0.4528	0.002	1000 Monte Carlo Trials
H_{02}	$\underline{S}$, 1.0000	0.00016	Approximation or 1000 MC Trials

We feel justified to proceed with testing H_{03} and H_{04}. We designate features 8 and 9 as categories separately, and we ask whether any one of the remaining features is unusually similar to these categories, based on statistics C_M and S_M. The critical level of C_M under H_{03} was estimated from 1000 Monte Carlo trials, while the critical level of S_M was obtained by both Monte Carlo means and analytical approximation. The results reported in Table 2.9 indicate that some feature is close to category feature 9 but no feature is unusually similar to category feature 8. Testing H_{04} produces the same result as testing H_{03} so only feature 9 is used as a category in further study.

Test results using the two statistics agree with each other. The approximate thresholds for S_8 and S_9 produce the same results as that derived by Monte Carlo means. These results support conclusions drawn ealier.

Table 2.9 Testing H_{03} and H_{04}

Hypothesis	Categorical feature	Statistic value		Cr. Level a^*	Method
H_{03}	8	C_8	0.1172	0.835	1000 MC Trials
H_{03}	9	C_9	0.2463	0.034	1000 MC Trials
H_{03}	8	S_8	0.9156	0.880	Approximation
H_{03}	8	S_8	0.9156	0.824	1000 MC Trials
H_{03}	9	S_9	0.9984	0.039	Approximation
H_{03}	9	S_9	0.9984	0.025	1000 MC Trials
H_{04}	8	C_8	0.1172	0.379	Exact
H_{04}	8	S_8	0.9156	0.379	Exact
H_{04}	9	C_9	0.2463	0.005	Exact
H_{04}	9	S_9	0.9984	0.005	Exact

2.5.2 Verification on Original Pattern Matrix

We now ask whether the grouping of patterns defined by a particular categorical feature could have been defined in a purely random manner. We measure similarity between patterns i and j by the Jaccard coefficient [1], J(i,j). The reason for using the Jaccard coefficient is that (1,1) matches are more important than (0,0) matches in questionnaire data. If n_{11} is the number of (1-1) matches in rows i and j of the pattern matrix and n_{00} is the number of (0-0) matches, then

$$J(i,j)=n_{11}/(M-1-n_{00}).$$

A category matrix is defined by:

B(i,j) = 1 if rows i and j have the same value for category feature M

= 0 else.

The test statistic, denoted by G_{amma}, is the point serial correlation between matrices [J(i,j)] and [B(i,j)]. The baseline is defined by H_{03}. A test of hypothesis based on G_{amma} is called a Mantel test [53,54] in other applications and tests whether the increases and decreases observed in the two matrices are unusually similar. The

distribution of G_{amma} was estimated by Monte Carlo means and the critical level was found to be 0.049 for the questionnaire data, using feature 9 as category. This suggests that H_{03} be rejected, which implies that the set of features is significantly related to category feature 9.

2.5.3 Feature Extraction Using Feature 9 as Categorical

Previous tests suggest that individual features and the entire matrix are unusually similar to categorical feature 9. Thus, we feel justified in seeking a subset of features which "explains" categorical feature 9 and which can lead to an efficient design for future questionnaires.

We chose a sequential forward stepwise selection method for study [86]. The best individual feature is chosen first. The best pair containing the best individual feature is then found, and the best triple containing the best pair is identified. This process is continued until a suitable level of performance is observed. Although only an exhaustive search of all subsets ensures optimality with binary features [8,9,14,15], the stepwise procedure is computationally attractive and is recommended in the literature [58,86]. The criterion used to compare two subsets of features is recognition rate under the

leave-one-out method. The five features selected, in order of selection, were features 10, 4, 6, 7, and 16. Feature 16 records whether a brother had cancer.

To verify these results, we repeated the Mantel test described above using only the five features selected from the questionnaire data. The critical level of the G_{amma} statistic was 0.003, which suggests that the patterns cluster by category unusually well and that the five features reflect category better than the set of all features.

2.5.4 Feature Extraction Using Feature 8 as Categorical

We carried out feature extraction and Mantel test using feature 8 as category. The critical level of G_{amma} for the entire matrix is 0.653, which suggests accepting H_{03}. Feature 1 had recognition rate 0.97, no lower than any other feature or any feature subset selected by our stepwise procedure. The critical level of G_{amma} using Feature 1 alone is 0.093. These results are summarized in Table 2.10.

Table 2.10 Critical Levels of Gammas

category	use all features	use selected features
8	0.653	0.093
9	0.049	0.003

Table 2.10 indicates that feature 8 is not a valid categorical feature. It might be caused by improper design of questionnaires. Its high recognition rate is due to the fact that there are only four 1's in column 8. Blindly performing feature selection and trusting in the recognition rate alone is misleading.

2.6 Summary and Conclusions

This chapter discussed the problem of preliminary analysis of dichotomous features. Four null hypotheses are stated to describe randomness in feature relationships. The necessity of this preliminary analysis is examined for the first time and is demonstrated with questionnaire data where much irrelevant information is usually involved. Rejecting these null hypotheses give us confidence in performing feature extraction. Accepting one of these null hypotheses prevents us from useless work and misleading results in feature extraction. This point is clear in the application

where preliminary analysis shows that feature 8 is not a valid categorical feature, although the feature extraction could give high recognition rate.

In testing H_{01} to H_{03}, statistics based on the correlation coefficient require Monte Carlo simulation to estimate thresholds. The statistics based on similarity S can be approximated analytically. Under H_{04}, both statistics, C_M and S_M, have known distribution, while S_M is distributed uniformly. For detecting Bahadur type relations between features, $\underline{S}$ and S_M have reasonable power. These facts suggest that $\underline{S}$ and S_M are better than $\underline{C}$ and C_M in preliminary feature analysis.

The Mantel test can also prevent us from accepting a misleading result, but estimating the critical level of G_{amma} requires either complicated asymptotic normal approximations [33,34] or Monte Carlo simulation. The test of H_{03} or H_{04} using S_M does not require Monte Carlo work and gives the same conclusion. This fact suggests that S_M is more suitable than G_{amma} for exploratory type analysis which requires speed and simplicity.

Tests of H_{01} and H_{02} using the two types of statistics give similar results for questionnaire data; tests of H_{03} and H_{04} also provide similar results. Since the thresholds of $\underline{S}$ and S_M are easier to compute in the cases of H_{02} and H_{04} than those for $\underline{C}$ and C_M, we may choose to limit ourselves to testing H_{02} and H_{04}. In this case, the following simplified methodology is proposed.

(1) Test H_{02} on the original pattern matrix using $\underline{S}$.
(2) If H_{02} is accepted, stop and try to gather more data.
(3) If H_{02} is rejected, choose categorical feature M.
(4) Test H_{04} on the pattern matrix with categorical feature M eliminated, using S_M.
(5) If H_{04} is accepted, stop and try other categorical features or gather more data.
(6) If H_{04} is rejected, go on with feature extraction.

CHAPTER 3

ADEQUACY OF BINARY PARTITIONS

This chapter studies procedures to assess the significance of a binary partition of a pattern set. For example, in cluster analysis one must verify structures resulting from clustering methods. Verification procedures assess the global fit of a hierarchy, the global fit of a partition, and the isolation and compactness of individual clusters. A sequence of partition fits, i.e., a sequence of partitional adequacy measures, can provide a basis for assessing global fit of a hierarchy [17]. We examine the application of the S statistic defined in Chapter 1 to two problems. One is the use of S as an external measure for binary partitions in cluster analysis. The second application is the design of a binary tree pattern classifier, where S measures the correspondence of a feature vector to category.

Section 3.1 compares S, under the permutation hypothesis defined in Chapter 1, to three other commonly used measures in a cluster validity framework. The relations among them are derived and their powers and computational costs are compared. Section 3.2 uses S in binary tree classifier design in comparison with the mutual information criterion [74]. A brief conclusion is given in Sec. 3.3.

3.1 Four External Measures of Association for Cluster Validity

A clustering algorithm begins with a measure of proximity between all pairs of objects in a set and induces a clustering, or partition, of the objects in which "dissimilar" objects are placed in separate clusters. Clustering algorithms will impose partitions even on totally random data, so a clustering must be validated to establish its "adequacy", or its degree of unusualness, when compared to a hypothesis of randomness [13].

Two types of measures of partitional adequacy are used in cluster analysis: external and internal [33,56]. An external criterion validates a partition with respect to a

source of information that is independent from that used to form this partition. Permutation statistics provide clear external criteria of partitional adequacy. For example, if distances between patterns were used to establish a partition, then category labels can be the second source, and the partitional adequacy measure answers the question: Do the patterns cluster by category? A prior conjecture, the result of another clustering algorithm, or some independent proximities could also serve to define the second partition. An internal criterion assesses the fit of a partition using only the proximity data from which the partition was generated. Several external criteria for partitional adequacy have been proposed [13,56].

The problem of partitional adequacy has been formulated under two hypotheses of randomness. The Random Graph Hypothesis [3] assumes that all possible rank order proximity matrices are equally likely. The permutation hypothesis [31] assumes that only those proximity matrices corresponding to a relabeling of patterns are equally likely. The permutation hypothesis provides a good baseline for assessing statistical significance [32,36,37]. In this section, we adopt the permutation hypothesis as the null hypothesis and compare S to three related measures, the Rand, G_{amma} and B statistics in terms of power and

computation time. Section 3.1.1 gives the definitions of the four statistics and Sec. 3.1.2 discusses the relation among them. Our goal is to examine the effectiveness of the statistic $S'=|S-0.5|$ in measuring the adequacy of binary partitions.

3.1.1 Definitions

Binary partitions are the basis for hierarchical clustering [1] and for binary tree classifier design. If we consider a hierarchy resulting from some hierarchical clustering method as a sequence of binary splits, then a binary partitional adequacy measure can be use to validate one split at a time and indicate from which point the splitting is random. In this section, we study four measures for the adequacy of binary partitions. These statistics assess the unusualness of a particular pair of partitions by testing the permutation hypothesis H_0 (Chapter 1). All have unimodal null distributions so a threshold can be selected under the null distribution for defining the significance of association.

Consider two independent binary partitions of a set of L patterns. Each pattern in each partition is coded as 0 or 1. The coding is presented as two binary L-vectors,

$V_1=[V_1(i)]$ and $V_2=[V_2(i)]$. Indicator matrices X and Y are defined as follows.

$$X(i,j)=1 \text{ if } V_1(i)=V_1(j),$$
$$=0 \text{ otherwise;}$$

$$Y(i,j)=1 \text{ if } V_2(i)=V_2(j),$$
$$=0 \text{ otherwise.}$$

The four cells in Table 3.1 contain all the information about the similarity between X and Y; $K=L(L-1)/2$.

Table 3.1 Frequencies of Combinations

		Y = 0	Y = 1	
X	0	a	b	$K-N_1$
	1	c	d	N_1
		$K-N_2$	N_2	K

For example,

$$a = |(i,j) : i<j, (X(i,j)=0) \cap (Y(i,j)=0)|.$$

The Rand statistic R [67], also called the simple matching coefficient, is the most commonly used measure of association between X and Y. It is also a basis of comparison for permutation statistics [17].

$$R = \frac{a + d}{K} = pd + q$$

where $p = 2/K$ and $q = (K-N_1-N_2)/K$

The G_{amma} statistic is the point serial correlation between the two indicator matrices and is used in Mantel tests [31,53,54].

$$G_{amma} = \frac{d - [(c+d)(b+d)/N]}{\sqrt{(a+b)(c+d)(b+d)(a+c)/\ N^2}} = rd + t$$

where $r = \frac{N}{\sqrt{N_1N_2(N-N_1)(N-N_2)}}$ and $t = N_1N_2$

Fowlkes and Mallows [17] recently proposed the B statistic as an external criterion for clustering.

$$B = \frac{d}{\sqrt{(c+d)(a+b)}} = zd$$

where $z = (N_1(K-N_1))^{.5}$

The distributions for the three statistics above do not have known analytical form under H_0 and must be estimated by Monte Carlo means or approximated by normal distributions. The fourth statistic is

$S' = |S(1,2)-0.5|$ where $S(i,j)$ is defined in Chapter 1.

The statistics R, B and G_{amma} can also be used as external criteria for partitions with an arbitrary number of components.

3.1.2 Relation Among Measures

We are interested in whether the four statistics S', R, G_{amma} and B are essentially the same in practice. First, we establish algebraic relations among them. We then study the sample correlation between S' and G_{amma} and compare their powers.

Under the permutation model and for fixed values of N_1 and N_2, the three statistics, R, G_{amma} and B are all linear functions of d. Therefore, they are linear functions of each other. In order to study the relation between S' and the other three statistics, we need only consider the

relation between S' and G_{amma}. Since G_{amma} is a linear function of d, it is sufficient to study the relation between S' and d. Under the permutation model, $d' = |\{i: V_1(i)V_2(i)=1\}|$ is the only independent variable. We investigate how d and S' vary with d'.

The S statistic is defined in Chapter 1 in terms of L, n_1, n_2 and (a',b',c',d'). When L, n_1 and n_2 are fixed, d' takes values in the fixed range: $[n_1+n_2-L,\ \min(n_1,n_2)]$. In this entire range, it is clear that E(S) under H_0 is a monotone increasing function of d'. Therefore, since S has uniform distribution under H_0, $E(S') = |E(S)-0.5|$ decreases first, reaches a minimum then increases as d' varies over this range. The quantity d is the following function of d'.

$$d = \binom{a'}{2} + \binom{b'}{2} + \binom{c'}{2} + \binom{d'}{2} = 2d'^2 + (L-2n_1-2n_2)d' + f$$

where $f = n_1^2+n_2^2+n''^2-n_1-n_2-n''$ and $n'' = L-n_1-n_2$

This function reaches its minimum when $d' = g = (2n_1+2n_2-L)/4$. If $n_1+n_2-L > g$, d and G_{amma} are monotonically increasing functions of d' over the entire range of d'. Thus the non-monotone function S' does not measure the same thing as G_{amma}. If $n_1+n_2-L < g$, such as in

the case $n_1=n_2=L/2$, both d and S' are non-monotone functions of d'. Figure 3.1 demonstrates how d and E(S') vary as functions of d' under H_0 when L=20, $n_1=9$ and $n_2=13$.

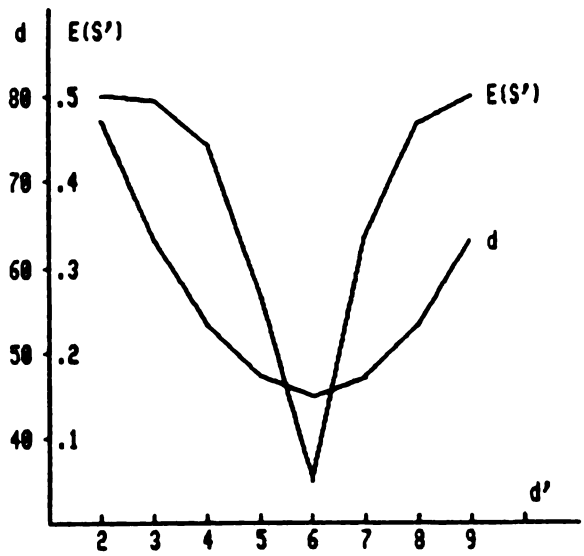


Fig.3.1 d and E(S') vs d' when L=28, $n_1=9$ and $n_2=13$

Table 3.2 gives some examples of the ranges of d' and g values. The cases marked with a star mark exhibit the general behaviour of Figure 3.1.

Table 3.2 Examples of d' Ranges and g Values

L	n_1	n_2	range of d'	g
10	3	3	(0, 3)	0.5 *
10	3	5	(0, 3)	1.5 *
10	3	6	(0, 3)	2.0 *
10	5	5	(0, 5)	2.5 *
10	5	6	(1, 5)	3.0 *
10	6	6	(2, 6)	3.5 *
20	5	5	(0, 5)	0.0
20	5	10	(0, 5)	2.5 *
20	5	13	(0, 5)	4.0 *
20	10	10	(0, 10)	5.0 *
20	10	13	(3, 10)	6.5 *
20	13	13	(6, 13)	8.0 *
40	10	10	(0, 10)	0.0
40	10	20	(0, 10)	5.0 *
40	10	25	(0, 10)	7.5 *
40	20	20	(0, 20)	10.0 *
40	20	25	(5, 20)	12.5 *
40	25	25	(10, 25)	15.0 *
80	20	20	(0, 20)	0.0
80	20	40	(0, 20)	10.0 *
80	20	50	(0, 20)	15.0 *
80	40	40	(0, 40)	20.0 *
80	40	50	(10, 40)	25.0 *
80	50	50	(20, 50)	30.0 *
200	50	50	(0, 50)	0.0
200	50	100	(0, 50)	25.0 *
200	50	130	(0, 50)	40.0 *
200	100	100	(0, 100)	50.0 *
200	100	130	(30, 100)	65.0 *
200	130	130	(60, 130)	80.0 *

The quantity g is inside the range of d' when n_1 and n_2 are around L/2. These cases exhibit the behavior in Figure 3.1. To further study the relation between S' and G_{amma}, we examine the sample correlation for the special cases when $n_1=n_2=L/2$, which is representative of the cases with star marks in Table 3.2. We generated vector

pairs of different lengths. For each permutation of one vector, we compute S' and G_{amma} which is converted to its z-score [33,34]. The correlation coefficients are shown in Table 3.3, which suggests that S' and G_{amma} are highly correlated under H_0 when the number of 0's and 1's are equal.

Table 3.3 Sample Correlation between S',E(S') and z-score of G_{amma}

no.of perms	vector size	Pearson corr(S',z)	Kendall's Tau corr(S',z)	Kendall's Tau corr(E(S'),z)
1000	30	0.873	0.900	1.000
1000	60	0.901	0.962	1.000
1000	90	0.913	0.969	1.000
10000	120	0.912	0.980	1.000

The sample space consists of all permutations of one vector. Under any alternative a non-uniform probability assignment is made to these permutations. The fact that the two statistics have correlation coefficient 1 under H_0 thus implies they have identical power under any alternative. Since R, B and G_{amma} are linear functions of each other, we conclude that they have the same power against any alternative hypothesis. When $n_1=n_2=L/2$, S' is highly correlated to R, G_{amma} and B. Since the rank order correlation between S' and G_{amma} is above 0.9, we expect

that they have compatible power against any alternatives. Since the threshold for S' can be determined in a natural fashion and it is comparable to other statistics, S' is a good choice for assessing partitional adequacy.

3.2 Binary Tree Classifier Design

In this section, we apply S' to the design of a binary tree classifier. Section 3.2.1 briefly discusses the concept of tree classifier, some existing techniques, and where this study fits. Section 3.2.2 describes the computation of feature thresholds from S. Section 3.2.3 considers the efficiency of computing feature thresholds. Some numerical examples are shown in Sec. 3.2.4.

3.2.1 Binary Tree Classifiers

Binary tree classifiers have been used in many pattern recognition problems [50,51,59,72,73,75,85,90]. Each node of the tree corresponds to a subset of features. A pattern to be classified is passed through the tree from the root to a leaf. At each non-terminal node, based on the subset of feature values corresponding to that node, the pattern is sent to one of two descendent nodes. Every terminal node, or leaf, is labeled by category. The unknown pattern

eventually reaches a leaf and is assigned the corresponding category label.

A tree classifier is designed from training patterns. At each non-terminal node, a subset of features is selected based on the training patterns available at the node. The training pattern set is divided into two disjoint subsets according to these features for use at successor nodes. A terminal node is labeled by the category of the majority of training patterns remaining. Since the computational cost in classification is roughly proportional to the square of the number of features used [81], we follow the literature and use one feature per node. Thus, descendent nodes are chosen by comparing the value of the feature corresponding to that node to a predefined threshold.

The advantages of a tree classification rule versus a single stage classification rule are in computational efficiency, use of features, avoidance of the "curse of dimensionality" and ease in human interpretation [2,38,50,59,65]. One disadvantage is that the design of a binary tree classifier often requires a large amount of computing time and storage [75], especially when a fully optimal tree is desired [65].

The design of a binary tree classifier with one feature per node consists of two components [46,49,75]: (1) Defining the structure of the tree; (2) Choosing the most effective feature and threshold at each node. Some design criteria [46,55,90] are low error rate, minimum number of nodes on the tree, shortest path length, and weighted sum of these factors [46]. Numerical examples show that local optimality does not ensure global optimality and that no simple method exists for specifying the optimal tree structure in a given problem [47]. Therefore, only conditional optimality can be achieved. Game tree search techniques and the look-ahead property have achieved partial global optimality [76]. Most practical tree designs use heuristic approaches and make no claim of optimality [50,59,70,73,75,81,90]. Our study provides an alternative heuristic approach, without optimization or look-ahead.

Systematic procedures have been developed for the first component [46,55,65]. Given thresholds for features which partition the feature space, Meisel's dynamic programming method will generate equivalent partitions of that space which are optimal in the sense of having min-max path length, minimum number of nodes and minimum expected path

length. Since the number of possible trees under the constraint of a given partition is still very large, this algorithm is not feasible for large numbers of features [55,65].

Many approaches have been proposed to attack both components of tree design simultaneously [48,67,74]. A specific criterion is selected to determine the feature thresholds. At each node, the best threshold of the best feature splits the training pattern set on this node. This splitting criterion is used at every non-terminal node until the tree is constructed. The maximum distance between the empirical c.d.f.'s of the feature under different categories [70] and the mutual information between the category and the thresholded feature [74] have been used as criteria. We examine a splitting criterion based on the S statistic between a thresholded feature vector and category vector for the two-class problem.

3.2.2 Computation of Feature Thresholds

Consider a non-terminal node with N training patterns, each being described by M features. The features must be at least on an ordinal scale, although binary features are allowed. The data are represented as an NxMx2 matrix A,

where A(i,j,1) denotes feature j of pattern i, A(i,j,2) denotes the category label of pattern i for all j, and A(*,j,1) and A(*,j,2) denote the feature and category vectors respectively for feature j. Order the entries so that for each j, A(i,j,1) becomes the ith smallest value and A(i,j,2) is the corresponding category label. A threshold value between A(i,j,1) and A(i+1,j,1) creates a binary vector A_i(*,j,1) with i 0's and (N-i) 1's. A similarity measure between A_i(*,j,1) and the category vector A(*,j,2) serves as the splitting criterion. Table 3.4 is an example of feature and category values for feature j. Threshold 5.0 will result in binary feature vector A_2(*,j,1)=001111, while the threshold 8.0 will give A_4(*,j,1)=000011.

Before sorting:						
A(i,j,2):	0	0	0	1	1	1
A(i,j,1):	8.21	7.43	1.20	5.56	4.23	8.02
After sorting:						
A(i,j,2):	0	1	1	0	1	0
A(i,j,1):	1.20	4.23	5.56	7.43	8.02	8.21

Table 3.4 Ordering of Training Patterns by Feature j (i=1,2,...,6)

The splitting of a node can be described as the Pascal-like procedure SPLIT written below.

```
Procedure SPLIT (node,MINCUT);
Begin
   Build matrix A for this node and set N,M;
   SMAX := 0;
   For j=1,M  begin
      Sort to obtain A(*,j,1);
      Arrange A(*,j,2) accordingly;
      For i=1,N-1 begin
         THRESH(j):=(A(i,j,1)+A(i+1,j,1))/2;
         For k=1,i A_i(k,j,1):=0;
         For k=i+1,N A_i(k,j,1):=1;
         S(i,j):=SIMILAR [A_i(*,j,1),A(*,j,2)];
         if S(i,j)>SMAX then begin
            SMAX := S(i,j); IMAX :=i; JMAX :=j
         end; (* if *)
      end; (* for *)
   end; (* for *)
   if SMAX > MINCUT then begin
      For i=1,IMAX the pattern corresponding to
         A(i,JMAX,1) is passed on to the left-son;
      For i=IMAX+1,N the pattern corresponding to
         A(i,JMAX,1) is passed on to the right-son;
      SPLIT(left-son,MINCUT); SPLIT(right-son,MINCUT)
   end; (* if *)
end; (* procedure *)
```

The algorithm for designing the whole tree is simply SPLIT(root,MINCUT), where MINCUT is the user specified minimum splitting criterion value. Several splitting criterion functions SIMILAR can be used, such as average mutual information [74]. We propose

S'=|S[(A(*,j,1),A(*,j,2)]-0.5|

as the SIMILAR function, since it has a direct intepretation and known distribution under H_0 (Chapter 1).

3.2.3 Efficiency in Feature Threshold Computation

Consider the N training patterns assigned to a node and feature j, represented by sorted arrays A(*,j,1) and A(*,j,2). There are at most (N-1) possible thresholds in the procedure SPLIT. To avoid checking every possible threshold, we investigate whether the best feature threshold occurs between runs of 0's or 1's in the A(*,j,2) vector, or occurs at the boundaries of the largest runs. Sethi [74] suggested restricting the search for feature thresholds to the boundaries of runs, i.e., check threshold [A(i,j,1)+A(i+1,j,1)]/2 only if $A(i,j,2) \neq A(i+1,j,2)$. Sethi didn't prove that other thresholds can be ignored when the mutual information is used as the splitting criterion.

Appendix A gives a simple induction proof that only thresholds at the boundaries of runs need be checked if S' is the splitting criterion.

We now ask whether threshold checking can be further restricted to the boundaries of the largest runs of 0's and 1's. To be specific, suppose the run lengths in A(*,j,2) are k_t,t=1,2,...,m and let

$$k^* = \max\{k_1, k_2, \ldots, k_m\}$$

which is achieved between pattern $k_1+\ldots+k_{i-1}+1$ and $k_1+\ldots+k_i$. We have observed that the threshold

$$[A(k_1+\ldots+k_{i-1},j,1)+A(k_1+\ldots+k_i,j,1)]/2 \quad \text{or}$$
$$[A(k_1+\ldots+k_i,j,1)+A(k_1+\ldots+k_{i+1}]/2$$

results in a binary feature vector with highest S' among all possible thresholds. We have not been able to prove this in general. Figure 3.2 demonstrates this phenomenon with 14 cases. In each case, two vectors are presented. The first vector is the category vector A(*,j,2). The second is the feature vector thresholded to have the maximum E(S') value. In every case, the "best" threshold point is at a boundary of a largest run. The category vectors in the last two pairs are identical, i.e., the "best" threshold is not unique. If we can only check the boundaries of largest

runs, we can further reduce the time for finding feature thresholds.

```
1110000100    11011100010110    0100111101001001101
0001111111    00000011111111    0000111111111111111

01011110    1101001111    0100001100    010011010111
00011111    0000001111    0000001111    000000000111

0000000000110    0100110000000000000    1000000100
0000000000111    0000001111111111111    0111111111

0111000110110110    01000110    01000010    01000010
0000000111111111    00000111    00111111    00000011
```

Fig.3.2 Fourteen Examples of Category-Feature Vector Pair

3.2.4 Numerical Examples

Since there are infinitely many different types of classification problems, a formal comparison of binary tree design algorithms is not feasible. Following the literature, we use several numerical examples to demonstrate the use of S' as a splitting criterion in tree design and indicate its advantages over the mutual information criterion in an informal manner. Our experiments are on four artificial data sets and two real data sets.

Data sets 1 through 4 are from a cluster process and are generated on a computer. Each cluster has about the same number of patterns. The number of clusters is a Poisson random variable. Patterns are distributed around cluster centers randomly chosen in a unit hypercube according to a Normal distribution with diagonal covariance matrix, all of whose diagonal elements equal sigma, the spread factor. The smaller sigma, the more distinct the clusters. We code the patterns in odd numbered clusters "1" and those in even numbered clusters "0" to serve as category labels. One half of the patterns in each cluster are used for training while the rest are for testing. The parameters actually used are shown in Table 3.5.

Data sets 5 and 6 are created from the Munson handprinted Fortran character set, containing 48 samples of each of four letters, namely "I", "M", "O" and "X". Each character is represented by an eight-dimensional pattern [12]. Data set 5 includes characters "I" and "M". Data set 6 contains all four characters with "I", "O" being one category and "M","X" being the other. The first half of each cluster (Data sets 1 through 4) and each alphabet (Data sets 5 and 6) are used for training.

Table 3.5 Parameters of Artificial Data Sets

Data set	no. of training patterns	no. of testing patterns	no. of features	no. of clusters	sigma
1	74	76	6	5	.09
2	72	73	15	6	.20
3	71	74	15	6	.30
4	74	76	9	4	1.0
5	48	48	8	(2)	-
6	96	96	8	(4)	-

The mutual information and S' are used as splitting criteria to construct two binary classification trees for each data set. The procedure SPLIT is used in all cases. We specify a minimum splitting criterion MINCUT in every case. When no threshold for any feature exceeds MINCUT,the node in question will be a leaf. Using the direct interpretation of S', we set MINCUT for S' at 0.475 for all data sets, which corresponds to a significance level of 0.05. That is, S'>0.475 means that the thresholded feature vector and the category vector are more closely related than 95% of the pairs formed by permuting one of these vectors. The mutual information levels are explained with the data sets.

The tree classifiers are summarized in Table 3.6 through Table 3.11. The trees for data sets 1 and 2 are given and written in prefix form to give a general feeling for the tree structure. The labels in "<>" ("a" or "b") are labels of leaf nodes. Other numbers are the features used. For example, 2(4<b><a>)<b> indicates the tree in Figure 3.3.

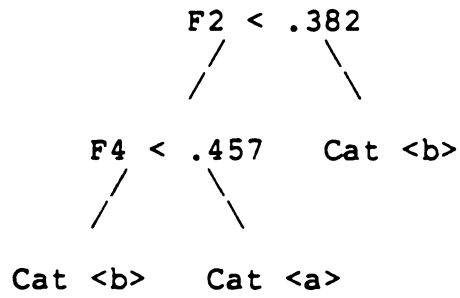


Fig.3.3 Tree Designed with S'
(minimum splitting S'=.475)
(artificial data set 1)

Table 3.6 Tree Design for Data Set 5

splitting criterion	mutual information	S'
MINCUT	0.330	0.475
number of nodes	3	
training recog.rate	1.000	
testing recog.rate	0.979	

Table 3.7 Tree Design for Data Set 2

splitting criterion	mutual information	S'
MINCUT	0.33	0.475
tree	6(<a>)(9(3(<b><a>)<b>)<b>)	
number of nodes	7	
training recog.rate	1.000	
testing recog.rate	0.918	

Table 3.8 Tree Design for Data Set 1

splitting criterion	mutual information	S'
MINCUT	0.33	0.475
tree	2(4<b><a>)<b>	4(<b>)(2<a><b>)
number of nodes	5	5
training recog.rate	1.000	1.000
testing recog.rate	0.987	0.987

Both methods gave the same result for Data set 5 (Table 3.6) and Data set 2 (Table 3.7). The difference between the trees for Data set 1 under the two criteria (Table 3.8) involves only the order of the features used and slight changes in feature threshold values. The minimum splitting criterion value MINCUT for mutual information was chosen because it gave good results for data set 5 in a preliminary trial. The distribution of mutual information is not known under H_0, although it has an asymptotic chi-square distribution [6]. These data sets suggest that the two criteria give comparable results.

Table 3.9 Tree Design for Data Set 3

splitting criterion	mutual information			S'
MINCUT	0.330	0.200	0.100	0.475
number of nodes	1	3	17	17
training recog.rate	0.563	0.761	1.000	1.000
testing recog.rate	0.568	0.730	0.716	0.716

Table 3.10 Tree Design for Data Set 4

splitting criterion	mutual information		S'
MINCUT	0.500	0.050	0.475
number of nodes	1	25	13
training recog.rate	0.635	1.000	0.865
testing recog.rate	0.618	0.566	0.566

Table 3.11 Tree Design for Data Set 6

splitting criterion	mutual information			S'
MINCUT	0.350	0.200	0.020	0.475
number of nodes	3	5	9	5
training recog.rate	0.885	0.979	1.000	0.979
testing recog.rate	0.885	0.938	0.917	0.938

Data set 3 (Table 3.9) is very loosely clustered. The recognition rates are lower than those for data sets 1 and 2. The value of MINCUT for mutual information has a significant effect on the tree obtained. Data set 4 (Table 3.10) is completely random. Different MINCUT values for the mutual information method result in very different trees. Data set 6 (Table 3.11) also shows the effect of MINCUT value with the mutual information criterion.

The above examples demonstrate that S' is a reasonable splitting criterion in tree classifier design. The trees designed with S' are as good as those designed with the mutual information criterion. Since S' has a known distribution under H_0 that is independent of the number of patterns at each node, S' has a direct meaning and the

MINCUT value for S' can be based on theory. The user must select the minimum threshold MINCUT under the mutual information criterion. These data sets demonstrate that the selection of MINCUT has a dramatic effect on the tree structure and on the recognition rates. Little prior information is available for selecting a mutual information threshold but a threshold on S' can be defined in a natural way. Mutual information has an asymptotic chi square distribution [6], but the degree of freedom depends on the number of patterns at each node. If one really wants the statistical significance to be consistent in the entire tree design, a p-value for mutual information can be approximated and different chi square tables can be used when the number of patterns changes from node to node, assuming the asymptotic distribution is applicable. This process is much more tedious than selecting the threshold of S'.

3.3 Summary and Conclusions

This chapter evaluated the similarity S, defined in Chapter 1, in two applications. Both applications require that the adequacy of a binary partition be measured. A version of S named S' is compared to three well known statistics (R, G_{amma}, B). The three are shown to be linear functions of each other. All have asymptotic normal

distribution under H_0. The measure S' is shown to be different from the others. When the numbers of 1's in both vectors are half the vector length, S' is highly correlated to other measures. The statistical significance of the other three measures demands either Monte Carlo simulation or rather complex approximation, while the significance level of S' is obvious.

We also applied S' to the design of a binary tree classifier. Numerical examples demonstrated that S' is a reasonable splitting criterion to be used in determining features and their thresholds. The known distribution and known statistical significance of S' permits one to establish the threshold of each feature in a simple and direct fashion. By comparison, the mutual information threshold must be approximated from asymptotic results and changes from node to node.

CHAPTER 4

TEMPLATE MATCHING

This chapter provides a probabilistic analysis of some statistics used in the template matching problem on binary images. We assume mathematical models for both null and alternative hypotheses. An approximately optimal statistic and two other statistics are derived and their powers are compared. We propose a suboptimal statistic which has reasonable power and is more sensitive to the true object location than existing statistics. Along with the experiments on artificial images generated under our mathematical model, this statistic is applied to several real Landsat-images.

The permutation statistic S defined in Chapter 1 and the Pearson correlation coefficient are also applied to matching binary images and are shown to provide result similar to other sub-optimal statistics. In this chapter, we concentrate on the statistic derived from the Neyman Pearson

Criterion, which is optimal under a weak assumption and which has reasonable power and sensitivity.

4.1 Introduction

Template Matching is a simple classical approach to the problem of locating an object in a digitized image [29,68,79,82]. An image is an N_r by N_c matrix of gray levels. A template is an $N_{r'}$ by $N_{c'}$ matrix containing a picture of an object with $N_{r'} << N_r$ and $N_{c'} << N_c$. The image may contain that object without rotation or size distortion, or it may contain no object at all. The object in the image is the same as the template except for lighting conditions. An example in Figure 4.1 shows an image and a template which are pictures of the same scene but are seen through different light frequency channels. This chapter considers only binary images and templates.

Fig.4.1 An Image and a Template

The standard solution to template matching is to scan through the image and compare the template with subimages at all possible template locations using some measure of similarity to decide whether or not the image contains an object and if it does, to estimate its location.

Since there are $(N_r-N_r,+1)(N_c-N_c,+1)$ locations for the template, the computation complexity of the standard solution to the template matching problem is

$$O((N_r-N_r,+1)(N_c-N_c,+1)(N_r,*N_c,)).$$

Several methods have been proposed to reduce the computational burden of the standard solution [5,43,60,61,66,87,88,89]. Bolles' planning method first applies an interest operator [57] to eliminate locations with low interest. This method is effective for grey level images and for images with structure, i.e. when pixels are dependent. Our study employs a randomness model with independent pixel values, which is appropriate for the Two Stage solution described below.

The first stage [83] compares a subtemplate to the subimages at all possible template locations. The second stage applies the entire template only at the locations with a sufficient match between the subtemplate and the subimage. The smaller the subtemplate, the lower the computational cost, but the higher the possibility of false match or missing a true match. The problem of choosing the optimal subtemplate size has been studied [83].

In one binary image model [83], a background pixel takes value 1 with probability p independent of other pixel values. The binomial distribution and the normal approximation describe the distributions of the simple matching coefficient between subtemplate and subimage. The independence assumption is more reasonable when the subtemplate is selected at random and is "sparse" [83].

In two stage matching, the simple matching coefficient, or equivalently, the absolute difference of pixel values between template and subimage, has low computational cost, but it has not been compared to other measures, such as the Jaccard coefficient and the correlation coefficient [29]. These measures have been compared in other applications

[80].

This chapter examines two stage template matching on a binary image and with a fixed binary subtemplate. We suggest a probabilistic model for both null hypothesis and alternative hypothesis and seek a powerful and computationally feasible statistic for recognizing the object. We do not attack the subtemplate size problem or the methodology of the second stage.

4.2 Mathematical Model

Section 4.2.1 discusses notation for the mathematical model. Section 4.2.2 defines five statistics for testing hypotheses. Approximations to the Neyman-Pearson statistic and two other statistics are defined in Sec. 4.2.3.

4.2.1 General Definition

An object is a mapping from a two dimensional grid of size $(N_r, xN_c,)$ to $\{0,1\}$. A template is a perfect copy of the object. In our experiment, each object pixel is generated independently with probability p' of being 1. A subset of n template pixels forms a subtemplate. Let q be the fraction of 1's in the subtemplate. An image G is a

matrix of size (N_rxN_c) with binary valued elements. The image may contain a single distorted version of the object (template). The template is scanned over the image but the image is seen only through the subtemplate. Let the possible locations of the template in the image be arbitrarily ordered from 1 to x_0. Let L_x denote the set of image pixels covered by a grid of size (N_r,xN_c,) at location x. Define $N_{ii}(x)$ as the number of pixels at which both the image and the subtemplate of a template at x are i for i=0,1, $0 \leq N_{ii}(x) \leq n$.

The mapping G is a matrix valued random variable and two hypotheses can be stated. We assume independent pixel values under both hypotheses. The template and the subtemplate are fixed.

H_0 : (No distorted object in the image),
Pr[G(i,j)=1|H_0]=p for any (i,j).

H_{1x}: (The distorted object is at location x),
Pr[G(i,j)=1|(i,j) not in L_x, H_{1x}]=p,
Let $T_x(i,j)$ be the template pixel value corresponding to image pixel (i,j) when the template is at location x. The parameters a and b measure distortion of the object in the image, such as lighting condition differences between the image and the template.

$$\Pr[G(i,j)=1 \mid (i,j) \text{ in } L_x,\ T_x(i,j)=0,\ H_{1x}]=a,$$
$$\Pr[G(i,j)=1 \mid (i,j) \text{ in } L_x,\ T_x(i,j)=1,\ H_{1x}]=b.$$

H_1 : (The distorted object is at some location),
{G(i,j)} are distributed as in H_{1x} for some x.

The random experiment consists of setting the template at all locations of the image and viewing the image at first only through the subtemplate. At each location y, $1 \leq y \leq x_0$, we observe $N_{00}(y)$ and $N_{11}(y)$. Let E_y denote the event that $N_{00}(y)= n_0(y)$ and $N_{11}(y)= n_1(y)$ for fixed numbers $n_0(y)$ and $n_1(y)$, where $n_0(y)+n_1(y) \leq n$.

Consider the likelihood ratio $R_{x''}$:

$$R_{x''} = \frac{\Pr(\bigcap_y E_y \mid H_{1x})}{\Pr(\bigcap_y E_y \mid H_0)}$$

Events $E_{y(1)}$ and $E_{y(2)}$ are not independent under H_0, H_1, or $\{H_{1x}, \text{ all } x\}$ when the templates at locations y(1) and y(2) overlap, but we treat $\{E_y\}$ as independent events to obtain an approximation to $R_{x''}$.

$$R_{x''} = R_{x'} = \prod_{y} \frac{\Pr(E_y|H_{1x})}{\Pr(E_y|H_0)} = \prod_{y} S_{xy}$$

If x=y, the template and the distorted object coincide under H_{1x}. In this case, $N_{00}(y)$ has a B[n(1-q),1-a] distribution and $N_{11}(y)$ has a B[nq,b] distribution, where B[M,p] is a binomial distribution representing the result of M independent Bernoulli trials with probability of success p. Under H_0, $N_{00}(y)$ has a B[n(1-q),1-p] distribution and $N_{11}(y)$ has a B[nq,p] distribution. In addition, $N_{00}(y)$ and $N_{11}(y)$ are independent under all hypotheses because the template and the subtemplate are fixed.

$$S_{xx} = \frac{\Pr(E_x|H_{1x})}{\Pr(E_x|H_0)} = u^{n_0(y)} v^{n_1(y)} C_{xx} \quad \text{where}$$

$$u = \frac{(1-a)p}{a(1-p)}, \quad v = \frac{b(1-p)}{(1-b)p},$$

$$C_{xx} = (a/p)^{n(1-q)}((1-b)/(1-p))^{nq}.$$

Similar equations can be stated for the special cases a=0, a=1, b=0, and b=1.

The term S_{xy} involves the distribution of the sum of two binomial random variables, so no closed form exists for S_{xy}. Poisson approximations create complicated expressions. We choose to use a binomial approximation, as explained in Appendix B. Let R_x denote the approximation to R_x, when S_{xy} is approximated as in Appendix B.

$$R_x = u^{n_0(x)} v^{n_1(x)} C_{xx} \prod_{y \neq x} \left\{ \left[\frac{w_{xy}}{(1-w_{xy})} \frac{p}{(1-p)} \right]^{n_0(y)-n_1(y)} C_{xy} \right\}$$

$$\log(R_x) = n_0(x)\log(u) + n_1(x)\log(v) + \log(C_{xx}) +$$

$$+ \sum_{---} \{[n_0(y)-n_1(y)]\log\left[\frac{w_{xy}}{(1-w_{xy})} \frac{p}{(1-p)}\right]\} + \sum_{---} \log(C_{xy})$$

4.2.2 Statistics for Testing H_0 vs. H_{1x}

To test H_{1x}, we simply place the template at position x and compute a statistic based on $N_{00}(x)$ and $N_{11}(x)$. The five statistics which we will compare are defined below.

$D_x = N_{00}(x) + N_{11}(x)$,

the simple matching coefficient,

$G_x = N_{00}(x)\log(u) + N_{11}(x)\log(v)$,

an approximation to the optimal statistic.

$J_x = N_{11}(x)/(n-N_{00}(x))$,
Jaccard coefficient, in which only (1,1) matches are considered, and n is the subtemplate size,

C_x = Pearson correlation coefficient,

S_x = the S statistic defined in Chapter 1.

The test using any of the above statistics is to reject H_0 when the observed value of the statistic is larger than some threshold. A Monte Carlo comparison study will be described in Sec. 4.3 to see if any of the statistics can match the performance of G_x.

4.2.3 Statistics for Testing H_0 vs. H_1

The most powerful statistic for testing H_0 vs H_1 can be obtained by combining the statistics for tests of H_0 vs H_{1x} and using the assumption that all positions are equally likely for the distorted object. Define

$$R'' = \frac{\Pr\left(\bigcap_{y=1}^{x_0} E_y|H_1\right)}{\Pr\left(\bigcap_{y=1}^{x_0} E_y|H_0\right)} = \frac{1}{x_0}\sum_{x=1}^{x_0} R_x''$$

We approximate each R_x'' by R_x from Sec. 4.2.1. The result is a sum of products.

$$R = \frac{1}{x_0}\sum_{x=1}^{x_0} R_x$$

The computational complexity of $N_{00}(y)$ and $N_{11}(y)$ at each location y is O(M). The complexity of computing R_x is $O(M\ k_y)$, where k_y is the number of positions in which templates overlap with the template at location y. Since k_y is of order (N_r', N_c'), the complexity of computing R is $O(x_0 M N_r', N_c')$. Since R uses the information provided in overlapping locations, it should be more powerful, but more complicated, than other statistics, such as D and G defined below, which have complexity $O(x_0 M)$. Statistics based on J_x, C_x and S_x are not considered since they act poorly for testing H_0 vs H_{1x} (Sec. 4.3).

$$D = \max_x(D_x)$$

$$G = \max_{x}(G_x)$$

Note that G_x uses only the information contained in $N_{00}(x)$ and $N_{11}(x)$ about location x and ignores the information in overlapping locations. The (0,0) and (1,1) matches are weighted according to parameters p,a and b. Statistic D is the special case of G that weights $N_{00}(x)$ and $N_{11}(x)$ equally, and is equivalent to the measure used elsewhere [69,83].

In order to compare D, G, and R in testing H_0 vs H_1, we define the sensitivities V(D), V(G), and V(R). Let x be the true object location,

$$V(D) = |\{ y : D_y \geq D_x \}|$$
$$V(G) = |\{ y : G_y \geq G_x \}|$$
$$V(R) = |\{ y : R_y \geq R_x \}|.$$

The sensitivity is the proper number of locations which should be identified for second stage template matching. The smaller this number, the better this statistic as a similarity measure in first stage template matching. Sensitivity is just as important as power for a criterion in first stage template matching.

4.3. Comparison Study

We compare the powers of D_x, J_x, C_x, S_x and G_x in testing H_0 vs H_{1x} for a fixed x. The image size and template size are both 8 by 8, the subtemplate size is 19 and pixels in the subtemplate are selected randomly. The four parameters (p,p',a and b) establish four factors whose levels are:

{0.2,0.4,0.6,0.8} for p and p',

{(0.1,0.9), (0.2,0.8)} for (a,b).

These parameters are known. In each case, 100 images generated under H_0 are used to establish a threshold and another group of 100 images under H_{1x} are used to estimate powers.

Based on the resulting power estimates, we performed a standard two sample t-test for each pair of comparison between statistics (Appendix C). The results indicate that D_x, J_x, C_x and S_x have essentially the same power, while D_x is slightly more powerful than the other three with significance level 0.6. Statistic G_x is more powerful than all others with at least 0.75 significance level. Therefore, we limit ourselves to G_x and D_x for testing H_0 vs

H_1.

Now we compare D,G and R in testing H_0 vs H_1 using the following criteria: (1) power, (2) sensitivity, (3) computational complexity, (4) feasibility of analytical thresholding. Our experiments study the effects of the following factors.

(1) Parameter values for p,p',a and b,

(2) State of knowledge: known or estimated parameters,

(3) Subtemplate size.

Statistics D and G are compared analytically, while a large scale Monte Carlo simulation is used to evaluate D,G and R. All tests have size 0.05. Empirical thresholds for tests are estimated from 100 Monte Carlo trials under H_0. Power estimates are based on 100 trials under the alternative hypothesis. The analytical comparison of D and G is given in Sec. 4.3.1. The Monte Carlo comparison of D, G and R is described in Sec. 4.3.2. Results are summarized in Section 4.4.

4.3.1 Analytical Comparison of D and G

The normal distribution is used to approximate the distributions of D and G under H_0 and H_1. We approximate powers for the following levels of parameters:

{0.2,0.5,0.8} for p and p',
{(0.1,0.9), (0.2,0.8)} for (a,b).

To obtain numerical values, typical image/template sizes of 16x16/8x8 and 32x32/12x12 were used.

Based on the resulting power estimates (Appendix D), we performed a standard two sample t-test which indicates that G is more powerful than D at level .025.

4.3.2 Monte Carlo Comparison of D, G and R

The image size is 16x16, the template size is 8x8 in this study. For each combination of the following parameter sets or variables, nine different random seeds are used to start the process. For every random seed, 100 trials are used to estimate the threshold and another 100 trials are used to estimate power. The parameters vary in the following ranges:

values of p: {.2,.5,.8};

values of p': {.2,.5,.8};

values of (a,b): {(.1,.9),(.2,.8)}.

There are five different states of parameter knowledge (coded in variable SPK). For SPK>0, p is estimated from 100 random images, different from the images used in estimating threshold and power.

SPK=0: know all parameters;

SPK=1: estimate p; p',a,b are known;

SPK=2: estimate p,p'; assume a=.10,b=.90;

SPK=3: estimate p,p'; assume a=.25,b=.96;

SPK=4: estimate p,p'; assume a=.05,b=.88.

Subtemplate sizes: {19,26,33,39};

The parameter p' is used only in statistic R. The subtemplate is selected randomly, except in one experiment (indicated in Table 4.5), in which the number of 0's and 1's are approximately equal.

4.4 Results

The experimental results for comparing D, G and R described above are collected in this section. In Sec. 4.4.1, Table 4.1 - Table 4.5 list the average powers of D, G and R. In Sec. 4.4.2, Table 4.6 - Table 4.10 list the average sensitivities of D, G and R. In Sec. 4.4.3, Tables 4.11 - 4.12 list the results of two sample t tests comparing powers and sensitivities of D, G and R.

4.4.1 Power Study Results:

Table 4.1 lists the mean powers P_D, P_G and P_R for the case when all parameters are known, the distortion is low, and the template size is relatively small. The marginals indicate the average effect of p and p'. From the overall total averages, we can compare powers between statistics and compare to results of other subtemplate sizes, other subtemplate selection methods, other parameter sets for (a,b), and other options for estimating parameters. In Table 4.1 and all following tables, "size" means the subtemplate size.

Table 4.1 Effect of (p,p') on Powers

SPK=0, (a,b)=(0.1,0.9), size=19

	p= .2	.5	.8	
P'=.2	.23 .31 .30	.80 .82 .95	1.00 1.00 1.00	.68 .71 .75
P'=.5	.89 .96 .99	.75 .75 .74	.83 .96 1.00	.82 .89 .91
P'=.8	1.00 1.00 1.00	.79 .82 .97	.25 .37 .33	.68 .73 .77
	.71 .76 .77	.78 .80 .89	.70 .78 .78	.73 .78 .81

P_D P_G P_R

Parameters p and p' indicate the difference in frequency of 1's between the background and the object. The further apart p and p', the easier the object is to detect. Our results confirm this. The mean powers of any statistic for the cases (p,p')=(.2,.2) and (.8,.8) are comparable, which reflects the symmetry in the problem. That is, reversing the coding should not alter the results. A similar comment can be made for other symmetric locations in the table. All three mean powers for the cases (p,p')=(.2,.2) and (.8,.8) are low, because the information for discrimination in these

cases is low. The expressions for G_x, u and v show that when p is small then u<<v so $N_{11}(y)$ is more heavily weighted than $N_{00}(y)$. That is, the discrimination is based primarily on $N_{11}(y)$, even though $N_{11}(y)$ is small. This weighting between $N_{00}(y)$ and $N_{11}(y)$ made G have higher power than D. Based on this understanding, we will develop a subtemplate selection method to increase $N_{11}(x)$ when both p and p' are small. The result of this method will be described in Table 4.5.

The first row of Table 4.2 contains the grand averages from Table 4.1. The second row contains the grand averages of a table for (a,b)=(0.2,0.8). Table 4.2 shows that increasing distortion between template and object in the image increases the difficulty in detection. Table 4.3 shows grand averages as functions of the state of parameter knowledge, while Table 4.4 shows them as functions of Subtemplate size.

Table 4.2 Effect of (a,b) on Powers
SPK=0, size=19.

	D	G	R
(a,b)=(.1,.9)	0.73	0.78	0.81
(.2,.8)	0.49	0.58	0.74

Table 4.3 Effect of Parameter Knowledge on Powers
(a,b)=(0.2,0.8), size=19

	D	G	R
SPK=0	0.49	0.58	0.74
1	0.49	0.63	0.74
2	0.49	0.58	0.68
3	0.49	0.56	0.69
4	0.49	0.57	0.67

Table 4.3 suggests that it is not crucial to know the distortion parameters exactly. Parameter p must be estimated from images similar to that used, or from prior knowledge about lighting conditions.

Table 4.4 Effect of Subtemplate Size on Powers
SPK=0, (a,b)=(0.1,0.9)

	D	G	R
subt.size= 9	0.39	0.41	0.70
13	0.56	0.60	0.74
19	0.73	0.78	0.81
26	0.83	0.87	0.87
33	0.88	0.94	0.93
39	0.92	0.96	0.95

Table 4.4 shows that increasing subtemplate size does,

indeed, improve the performance, which is intuitively expected. Since the optimal size problem has been solved [83], we did not study this in much detail, although the optimality was defined without an explicit alternative hypothesis.

Recall that when (p,p')=(.2,.2) or (.8,.8), the more heavily weighted number of pairs (N_{00} or N_{11}) is usually small when the subtemplate is selected at random. We applied another algorithm which selects a subtemplate with almost equal numbers of 0's and 1's. The results are shown in Table 4.5. Comparing Table 4.1 and Table 4.5 shows that the way of selecting the subtemplate increases power significantly when p=p' is very small or very large and does not affect it in other cases. Thus, a balanced subtemplate performs better than a randomly selected one. Actually, we might vary the fraction of 1's in the subtemplate according to the parameters p, p', a and b to get optimal results.

Table 4.5 Powers when Subtemplate is Balanced

	p= 0.2	0.5	0.8	
P'=.2	.79 .93 .92	.75 .76 .95	.97 1.00 1.00	.84 .90 .96
P'=.5	.89 .96 .99	.71 .72 .71	.92 .99 1.00	.84 .89 .90
P'=.8	.98 1.00 1.00	.78 .76 .97	.83 .94 .94	.86 .90 .97
	.89 .96 .97	.75 .75 .88	.91 .98 .98	.85 .90 .94

P_D	P_G	P_R

4 - 4 _2 Sensitivity Study Results

Tables 4.6 - 4.10 show the results of the sensitivity study for the conditions of Table 4.1 - 4.5, respectively.

Table 4.6 Effect of (p,p') on Sensitivities
SPK=0, (a,b)=(0.1,0.9), size=19

	p= 0.2	0.5	0.8	
P'=.2	6.22 4.01 4.01	2.10 2.06 2.42	1.24 1.26 1.65	3.19 2.44 2.70
P'=.5	1.22 1.13 1.24	1.23 1.19 1.19	1.22 1.12 1.30	1.22 1.15 1.24
P'=.8	1.23 1.27 1.97	2.08 1.94 2.53	6.80 3.63 3.63	3.37 2.28 2.71
	2.89 2.14 2.40	1.80 1.73 2.05	3.09 2.00 2.19	2.59 1.96 2.21

V(D)
V(G)
V(R)

Table 4.7 Effect of (a,b) on Sensitivities
SPK=0, size=19

	D	G	R
(a,b)=(.1,.9)	2.59	1.96	2.21
(.2,.8)	7.44	4.98	5.22

Table 4.8 Effect of Parameter Knowledge on Sensitivity
(a,b)=(0.2,0.8), size=19

	D	G	R
SPK=0	7.44	4.98	5.22
1	7.21	5.01	5.14
2	7.27	4.47	5.15
3	7.44	6.41	7.53
4	7.44	4.94	5.58

Table 4.9 Effect of Subtemplate Size on Sensitivity
SPK=0, (a,b)=(0.1,0.9)

	D	G	R
subt.size= 9	8.29	7.03	6.06
13	4.88	3.81	3.69
19	2.59	1.96	2.21
26	1.61	1.27	1.50
33	1.25	1.09	1.28
39	1.10	1.02	1.19

Table 4.10 Sensitivities when Subtemplate is Balanced

	p= 0.2	0.5	0.8	
P'=.2	1.20 1.04 1.04	1.26 1.24 1.39	1.13 1.29 1.52	1.20 1.19 1.32
P'=.5	1.17 1.10 1.22	1.21 1.18 1.18	1.14 1.09 1.34	1.17 1.12 1.25
P'=.8	1.13 1.32 1.57	1.21 1.15 1.34	1.18 1.02 1.02	1.17 1.16 1.31
	1.17 1.15 1.28	1.23 1.19 1.30	1.15 1.13 1.29	1.18 1.16 1.29

V(D)
V(G)
V(R)

Tables 4.6 - 4.10 indicate that the effects of (p,p'),(a,b) and subtemplate size on sensitivity are similar to their effects on power. Estimating parameters does not affect sensitivity very much. Interestingly enough, statistic G is more sensitive than statistic R, except when the subtemplate is very small. An intuitive reason for this is that the high value of R_x may happen at several locations overlapping with the the true object location. The use of information at overlapping locations provides higher power than ignoring the overlap, but some sensitivity is lost.

4.4.3 Formal Comparison

Table 4.1 - 4.10 use only mean values of power estimates and sensitivity estimates and give a general idea of the effects of parameters. To compare the three statistics, we performed a standard two sample t test in each case of different subtemplate size. For example, we test the hypothesis Power(D)>Power(G). A positive t value indicates the acceptance of the hypothesis at the critical level w_t. A negative t value indicates the acceptance of the reverse hypothesis Power(D)<Power(G) at the critical level w_t. Tables 4.11 and 4.12 show the values of t and the values of w_t. The sample size for every block is 81. In Table 4.11, PWD, PWG and PWR represent powers of D, G and R respectively in tests of H_0 vs H_1.

Table 4.11 Power Comparison
(results of two sample t test)
SPK=0, (a,b)=(0.1,0.9)

size	PWD>PWG	PWD>PWR	PWG>PWR
9	-0.386 .35	-5.205 .00	-4.642 .00
13	-0.708 .24	-3.501 .00	-2.768 .00
19	-1.082 .14	-1.699 .05	-0.672 .25
26	-0.994 .16	-1.092 .14	-0.120 .45
33	-1.672 .05	-1.438 .08	0.280 .39
39	-1.958 .03	-1.450 .08	0.601 .27

t w_t

Table 4.12 Sensitivity Comparison
Result of Two Sample t Test
SPK=0, (a,b)=(0.1,0.9)

size	V(D)>V(G)	V(D)>V(R)	V(G)>V(R)
9	1.132	1.965	0.881
	.13	.03	.19
13	1.654	1.826	0.221
	.05	.04	.41
19	1.681	0.983	-0.881
	.05	.16	.19
26	2.098	0.649	-2.259
	.02	.26	.01
33	2.108	-0.335	-3.609
	.02	.37	.00
39	2.207	-1.665	-4.316
	.02	.05	.00

t
w_t

Table 4.11 indicates that R is more powerful than the other statistics when the subtemplate is small. This may be explained by the fact that overlapping locations provide most of the information for discrimination. Both R and G are significantly more powerful than D in almost all cases. When the subtemplate size is 19 or larger, R is only slightly more powerful than G. A similar situation exists for sensitivity, except that G is significantly more sensitive than R and D when subtemplate size is large. The

reason for R to perform worse when the subtemplate size is large might be the fact that some assumptions made in deriving R are violated when the subtemplate points are more dependent.

4.4.4 Results on Landsat Images

This study involves landsat images of the same scene from different light frequency channels. Each image was converted into a binary image using the average grey level of that image as threshold. We arrange the study into several cases. In each case, we take a subimage of size 8x8 from a channel i image to form an object or template. Then we take another subimage of size 64x64 from channel j ($i \neq j$) which includes the object. We consider that this 64x64 image contains a intensity distorted (not geometrically distorted) object. We apply our scheme to find the relative location of the distorted object. We studied five measures of similarity and three subtemplate sizes. Each study was repeated three times. The parameter p is estimated by the fraction of 1's in the 64x64 image from channel j. The parameter p' is estimated by the fraction of 1's in the 8x8 template. The distortion parameters (a,b) are assumed to be (0.1,0.9). A balanced subtemplate is formed in each case.

Table 4.13 Results on Several Landsat Images

subtemplate size	V(D)	V(G)	V(R)	V(max(C_x))	V(max(S_x))
9	1512	1211	718	2244	1222
9	1719	1456	829	2159	1188
9	154	110	1528	110	110
19	18	14	541	18	22
19	165	64	1630	42	36
19	1188	979	711	1117	1117
29	79	64	544	78	102
29	926	756	750	787	722
29	719	468	681	969	719

Among the nine cases in Table 4.13, G is more sensitive than D in every case; G is more sensitive than R in five cases; more sensitive than max(C_x) in seven cases; more sensitive than max(S_x) in five cases. These results support our conclusion about G drawn from artificial data.

In two cases using very small subtemplates, statistic R performs best of all statistics. However, in four cases using large subtemplates, R performs worst. The reason may be that the independent assumption we made in the derivation of R is violated for the Landsat images. Since R is the only statistic using the information from overlapping locations, it performs rather differently from all other statistics.

4.5 Summary and Conclusions

This chapter stated a null hypothesis (no object) and alternative hypotheses (single object present) for a problem in binary template matching and examined an approximation to the Neyman Pearson statistic for testing them. This statistic is compared to the simple matching coefficient and other similarity measures including the S statistic defined in Chapter 1. A new sensitivity index is proposed to assess the ability of a statistic to locate the true object in the image. Power, sensitivity and computation cost are the criteria for comparison.

In testing H_0 vs H_{1x}, G is more powerful than all other statistics. In testing H_0 vs H_1, R is most powerful, but less sensitive than G to the true object location. The statistic G, with the information in overlapping locations ignored, is almost as powerful as R, but computationally much simpler, and more sensitive to the true object location when the subtemplate is large. The threshold of G for testing H_0 vs H_1 can be obtained analytically, which is another important advantage of G over R.

The simple matching coefficient D, is less powerful and less sensitive than G, but has the same order of computational complexity as G. The correlation coefficient has been used for grey level image template matching problems [29]. Correlation and the S statistic defined in Chapter 1 have lower power and lower sensitivity than G and R, with computational complexity at least as large as G. Therefore, we suggest using G in the first stage of template matching.

CHAPTER 5

COMPUTATIONAL CONSIDERATIONS

Since the S statistic is computed directly from the cumulative hypergeometric distribution function (c.d.f.), we will discuss algorithms for computing hypergeometric c.d.f.'s that have appeared in the literature. Notation is defined in Sec.5.1. The Peizer approximation is described in Sec.5.2. We give a hardware architecture design for the recursive algorithm in Sec.5.3 and an overall computational time comparison in Sec.5.4. A brief conclusion is given in Sec.5.5.

5.1 Notation

Consider two binary N vectors V_1 and V_2. We first define n,r and a as follows, to be consistent with the literature [52].

Table 5.1 Observables in Vector Pair

V_1 \ V_2	1	0	
1	a	b	n
0	c	d	m
	r	s	N

For example, for the following vector pair,

$$V_1 = [\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ ]$$
$$V_2 = [\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ ]$$

N=8, n=3, r=4 and a=1.

For computational convenience [52], without loss of generality, we can recode V_1, V_2 and 0,1, so that

$$a \leq d \text{ and } a < b \leq c,$$

or, equivalently,

$$2a + 1 \leq n \leq r \leq m.$$

Also, we let k=min(a,b-1,c-1,d) [52].

The probability density function (p.d.f.) h(i) and the c.d.f. H(a) of the hypergeometric distribution are defined below.

$$h(i) = h(N,n,r,i) = \frac{\binom{n}{i}\binom{N-n}{r-i}}{\binom{N}{r}} \quad \text{for } 0 \le i \le a,$$

$$H(a) = \sum_{i=0}^{a} h(i) \quad \text{for } 0 \le a \le \min\{n,r\}.$$

The S statistic studied in the previous chapters can be written in the form of H(.) as follows.

$S(V_1,V_2) = H(a) - [H(a)-H(a-1)]U$

where U is an independent random variable distributed uniformly over [0,1].

The remainer of this chapter is concerned with various ways of computing H(a). The computation of the c.d.f. H(a) has attracted great attention. We compare them in terms of computation time, i.e. the number of time cycles involed in the computation. We first define the following notation.

T_a: number of cycles needed in addition,

T_m: number of cycles needed in multiplication,

T_d: number of cycles needed in division,

T_r: number of cycles needed in square root operation,

T_g: number of cycles needed in logarithm operation,

T_t: number of cycles needed in looking up a standard normal distribution table.

Each number is the time needed for each operation when a single CPU is used. It is also the number of segments in each functional pipeline unit; e.g., T_a is the number of segments in an pipeline adder. The computation time for each algorithm discussed below assumes a single CPU unless otherwise specified.

The order of operations is important since the word length in a computer is limited. The ratio of two long products should be done by alternating division and multiplication. Since the direct computation involves factorials and a great deal of repetition, it is very time consuming. The computation time is

$$aT_a + (a+1)\,[(2r-1)T_m + 2rT_d]$$

The following formula provides a recursive way to compute h(j).

$$h(0) = \frac{m(m-1)\ldots(m-r+1)}{N(N-1)\ldots(N-r+1)}$$

$$h(j+1) = h(j)X(j)$$

$$\text{where } X(j) = \frac{(n-j)(r-j)}{(j+1)(N-n-r+j+1)}$$

The computation time for the recursive formula is

$$(r-1)T_m + rT_d + a(2T_m + 2T_d + T_a).$$

5.2 Peizer Approximation

The approximate computation of H(a) has been investigated by many researchers and a recent extensive empirical study [52] of the accuracy of 12 normal and three binomial approximations showed that a normal approximation by Peizer is both far superior to other normal appoximations and simple to compute. Since the binomial tail is almost as difficult to compute as the hypergeometric tail, the binomial approximation is not recommended [52]. In this section, we briefly state the Peizer formula and give a simple discussion on computational complexity.

The Peizer approximation is

$$H(a) \sim F_n(z)$$

where F_n is the c.d.f. of the standard normal distribution,

and

$$z = \frac{A''D'' - B''C''}{|AD - BC|} \sqrt{\frac{2mnrsN'L}{m'n'r's'N}}$$

where $A = a+.5$, $A' = A+(1/6)$, $A'' = A'+\frac{.02}{A+.5}+\frac{.01}{n+1}+\frac{.01}{r+1}$,

B,B',B",C,... are defined in a similar manner from b,c,...,

and $L = A\log(AN/nr)+B\log(BN/ns)+C\log(CN/mr)+D\log(DN/ms)$.

The maximum absolute error of this approximation is .001 if $k>2$, with the maximum relative (percent) error being 0.71% through 19.7%. The guaranteed number of correct decimal places in this case is at least 3.040 [52]. When k is small, approximations are not needed since the exact computation is trivial.

The computation time of Peizer's approximation T(Peizer) is a constant, independent from the vectors V_1 and V_2. The computation time (using a single CPU) is

$$37T_a + 22T_m + 18T_d + 4T_g + T_r + T_t$$

We noticed that some portions of the computation can be done concurrently, e.g., $A\log(AN/nr)$ and $B\log(BN/ns)$. If we have sufficient hardware and perform all possible concurrent computations, the computation time will be reduced to

$$3T_a + 4T_m + T_d + T_g + T_r + T_t.$$

5.3 A Hardware Implementation

In recent years, considerable efforts have been devoted to developing special computer architectures for pattern recognition and image processing [20,63]. We desribe a two level pipeline design for the recursive computation of H(a). Section 5.3.1 is an overview of the architecture. A detailed discussion of the pipeline structure is given in Sec.5.3.2.

5.3.1 An Overview of the Architecture

The overall structure is shown in Figure 5.1. The input is the observables of vector pair V_1 and V_2. Since the computation of h(0) can be implemented by a straightforward design or by a table look-up, h(0) is considered as input in our design.

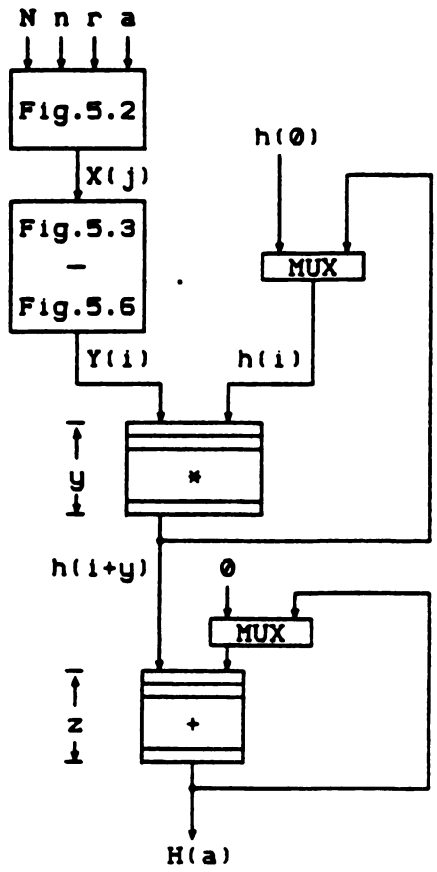


Fig.5.1 Overall Architecture for Computing H(a)

The bottom pipeline adder contains $z(=T_a)$ segments. The pipeline multiplier with $y(=T_m)$ segments computes values of h(.) in the following fashion.

$$h(i+y) = h(i)Y(i) \text{ where } Y(i) = X(i)X(i+1)...X(i+y-1)$$

The box on the upper left corner in Figure 5.1 produces X(j) for different j every cycle. The box right below it computes Y(i) from X(j), which is the main part of this design. The details of those two boxes are shown in Figures 5.2 through 5.6. The boxes marked "MUX" indicates a multiplexer and the small boxes with a decrement or increment input are counters. The input from above to these counters are initiation lines.

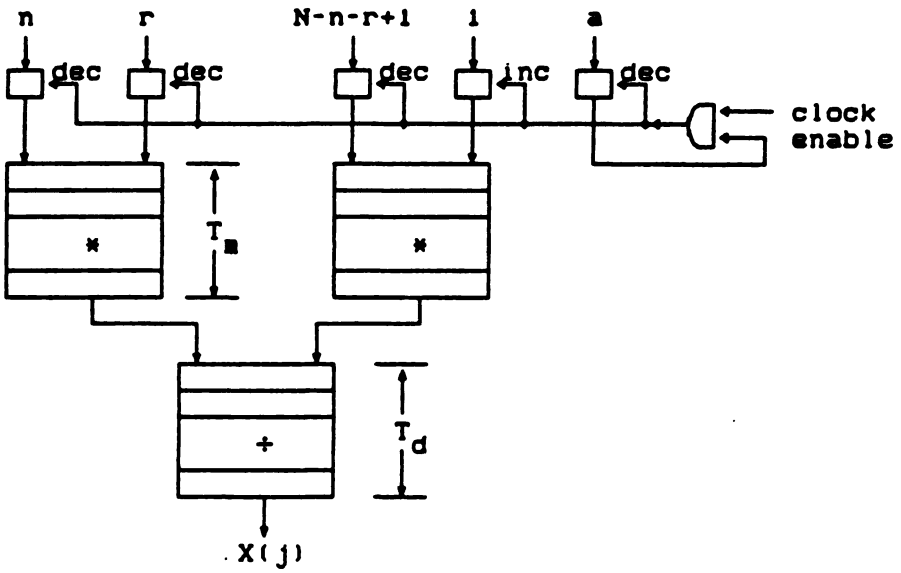


Fig.5.2 Compute X(j)

5.3.2 Computing Y(i)

We discuss this part of architecture in a general environment. The function is

$$Y(i) = X(i) * X(i+1) * \ldots * X(i+y) = \prod_{j=i}^{i+y} X(i)$$

where "*" can be any operator which is commutative and associative.

In this design, a pipeline functional unit with x segments is used and represented by a box marked with a letter "x". For the hypergeometric case, $x=y=T_m$. The design of the pipeline can vary with y. We consider y = 2,3,...,15, which are the usual lengths of pipeline functional units in computers such as Cray 1. Figures 5.3 through 5.5 show designs for y=4, 7 and 13. The general design is shown in Figure 5.6. In the cases y=4 and y=13, the design is the same as in the general case. The case y=7 takes advantages of individual y value and thus is different from the general design. A box with one input, one output and a number NUM, represents a NUM-bit shift register. Some of those boxes are small, with NUM=1,2,4. Some are larger with NUM=x,x-1,x-2.

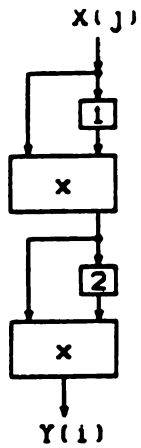


Fig.5.3 Compute Y(i) from X(j) when y=4

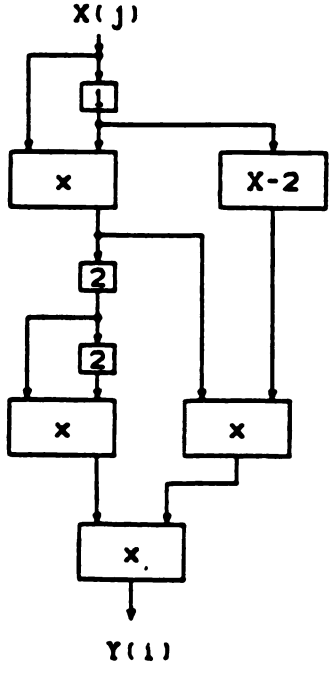


Fig.5.4 Compute Y(i) from X(j) when y=7

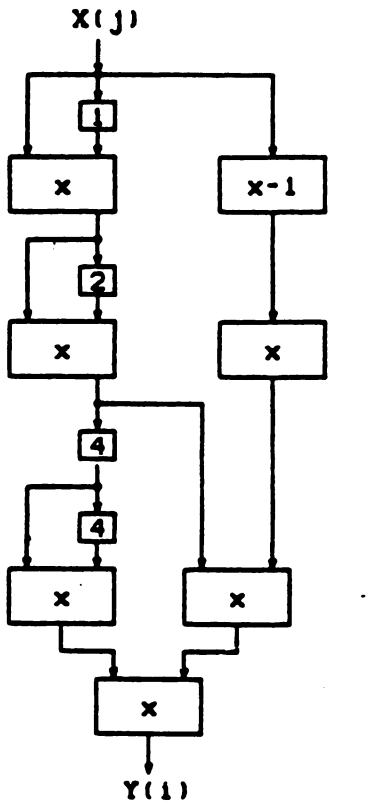


Fig.5.5 Compute Y(i) from X(j) when y=13

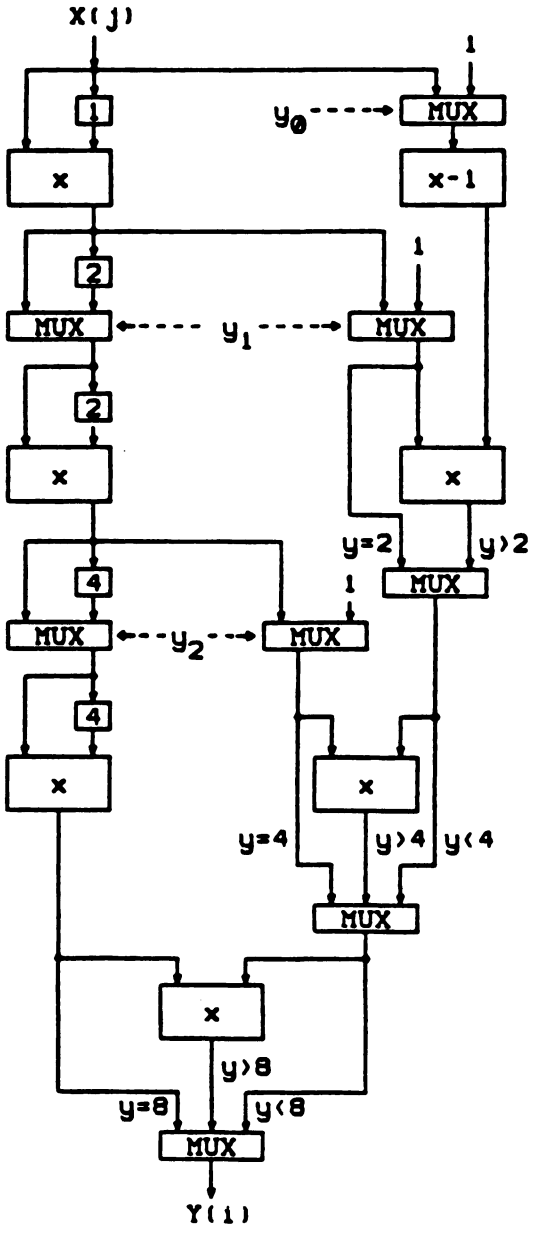


Fig.5.6 Compute Y(i) from X(j) (general design)

Multiplexer control is indicated either as conditions at input lines, or as a dash line at the side connected to the control signal. The control signals happen to be the bits in the binary expression of y.

$$y = y_3 y_2 y_1 y_0$$

To demonstrate how these pipelines work, we present a time diagram showing the data contents in different portions of the pipeline at different time steps. We pick y=7 and x=3 and give the pipeline design in Figure 5.7. Boxes (C1,C2,C3), (F1,F2,F3) and (I1,I2,I3) are pipeline units. Boxes B, D1, D2, E1, E2 and G are one-bit shift registers. The time diagram is shown in Figure 5.8.

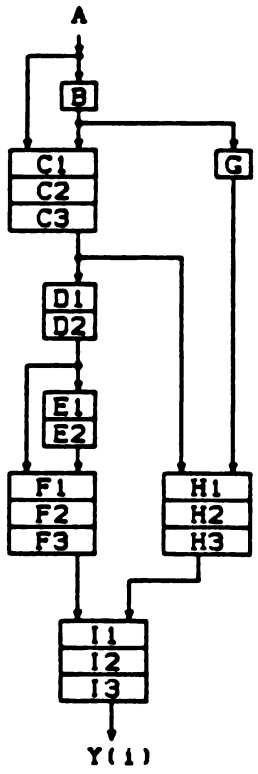


Fig.5.7 Compute Y(i) from X(j) when y=7 and x=3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
B		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C1		0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C3				0	0	1	2	3	4	5	6	7	8	9	10	11	12	13
					1	2	3	4	5	6	7	8	9	10	11	12	13	14
D1					0	0	1	2	3	4	5	6	7	8	9	10	11	12
						1	2	3	4	5	6	7	8	9	10	11	12	13
D2						0	0	1	2	3	4	5	6	7	8	9	10	11
							1	2	3	4	5	6	7	8	9	10	11	12
E1							0	0	1	2	3	4	5	6	7	8	9	10
								1	2	3	4	5	6	7	8	9	10	11
E2								0	0	1	2	3	4	5	6	7	8	9
									1	2	3	4	5	6	7	8	9	10
F1							0	0	0	0	1	2	3	4	5	6	7	8
								1	2	3	4	5	6	7	8	9	10	11
F3									0	0	0	0	1	2	3	4	5	6
										1	2	3	4	5	6	7	8	9
G			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H1				0	0	0	1	2	3	4	5	6	7	8	9	10	11	12
					1	2	3	4	5	6	7	8	9	10	11	12	13	14
H3						0	0	0	1	2	3	4	5	6	7	8	9	10
							1	2	3	4	5	6	7	8	9	10	11	12
I1							0	0	0	0	0	0	0	1	2	3	4	5
								1	2	3	4	5	6	7	8	9	10	11
I3									0	0	0	0	0	0	0	1	2	3
										1	2	3	4	5	6	7	8	9

Fig.5.8 Time Diagram of Figure 5.7

(The left most column is a circuit element, the numbers on the right are the contents in that element.)

The computation time and the circuit complexity of the part computing Y(i) from X(j) are shown in Table 5.2. The time complexity, denoted by T_y, is the number of time cycles through the pipeline. The circuit complexity, denoted by M_y, is the number of functional units used in the general design in Fig.5.6. Also, we define the following functions.

$f_1(y)$ = 1 if y is an odd number,
= 0 otherwise;

$f_2(y,a)$ = 1 if y > a,
0 otherwise.

Table 5.2 Time and Circuit Complexity of the Part Computing Y(i) from X(j)

	general design for all $y=2,3,\ldots,15$	individual design for each y value when $x>3$
M_y	$a\lfloor\log_2 y\rfloor = 6$	$\lfloor\log_2 y\rfloor + (y_3+y_2+y_1+y_0-1)$
T_y	$x\lceil\log y\rceil_2+y-1-f_1(y)$	
T_2	$x + 1$	$x + 1$
T_3	$2x + 1$	$2x + 1$
T_4	$2x + 3$	$2x + 3$
T_5	$3x + 3$	$3x + 3$
T_6	$3x + 5$	$3x + 3$
T_7	$3x + 5$	$3x + 5$
T_8	$3x + 7$	$3x + 7$
T_9	$4x + 7$	$4x + 7$
T_{10}	$4x + 9$	$4x + 7$
T_{11}	$4x + 9$	$4x + 7$
T_{12}	$4x + 11$	$4x + 7$
T_{13}	$4x + 11$	$4x + 11$
T_{14}	$4x + 13$	$4x + 11$
T_{15}	$4x + 13$	$4x + 13$

Table 5.2 demonstrated that the individual design for a fixed y value can be more efficient than the general design for $y<15$.

We now derive the total computing time of H(a), for Fig.5.1, which is denoted by T_{total}. We first define $T_{X(.)}$ to be the time for computing X(0) (Fig.5.2), define T_{merge} to be the time for the bottom pipeline adder to produce final result after h(a) is fed in the adder. We have

$$T_{total} = T_{X(.)} + T_y + T_m + a + 1 + T_{merge}.$$

$$T_{X(.)} = T_m + T_d$$

$$T_y = T_m\lceil \log_2 T_m \rceil + T_m - 1 - f_1(T_m)$$

$$T_{merge} = T_m\lceil \log_2 T_m \rceil - (2^{\lfloor \log_2 T_m \rfloor} - T_m) + f_2(T_m, a)(T_m - a)\lceil \log_2 a \rceil$$

$$T_{total} = (2\lceil \log_2 T_m \rceil + 4)T_m - 2^{\lfloor \log_2 T_m \rfloor} + T_d + a + f_2(T_m, a)(T_m - a)\lceil \log_2 a \rceil - f_1(T_m)$$

5.4 Comparison

We compare the computation times of all methods in Table 5.3. The word "soft" in parentheses means software serial computation on a single CPU; the word "hard" means special hardware which performs concurrent operation with possible pipeline structures. Since the actual computation in a computer depends on programming and the individual machine, the numbers in Tables 5.3 and 5.4 provide only the order of magnitude. The logarithm operation is performed by table look-up, all of which take 4 time cycles. The square root operation assumes a standard algorithm [39]. Based on Cray-1 parameters, every cycle is 12.5 nsec and the number of cycles for various operations are as follows.

$T_a = 6$, $T_m = 7$, $T_d = 29$,
$T_g = 4$, $T_r = 30$, $T_t = 4$.

The results in Table 5.3 are computed according to [64]. In Table 5.4, we show how the computing time ranges over values of a for various methods when r=50, 150 and 450. The times are in number of cycles. All the times are computed when Cray-1 parameters are used [10].

Table 5.3 Computation Time Comparison

	general form of computing time	typical computing time based on Cray 1
direct (soft)	$aT_a+(a+1)[(2r-1)T_m+2rT_d]$	72ar+72r-a-7
recursive (soft)	$(r-1)T_m+rT_d+a(2T_m+2T_d+T_a)$	36r+78a-7
recursive (hard)	$(2\lceil\log_2 T_m\rceil+4)T_m-2^{\lfloor\log_2 T_m\rfloor}+T_d+a+$ $+f_2(T_m,a)(T_m-a)\lceil\log_2 a\rceil-f_1(T_m)$	a + 94 + $+f_2(7,a)(7-a)\lceil\log_2 a\rceil$
Peizer (soft)	$37T_a+22T_m+18T_d+4T_g+T_r+T_t$	1026
Peizer (hard)	$3T_a+4T_m+T_d+T_g+T_r+T_t$	113

Table 5.4 Typical Computation Time Ranges
(for a=0,1,2,...,r)

	r=50	r=150	r=450
direct (soft)	3,593 ~ 183,543	10,793 ~ 1,630,643	32,393 ~ 6,223,335
recursive (soft)	1,793 ~ 5,693	5,393 ~ 17,093	16,193 ~ 51,293
recursive (hard)	94 ~ 144	94 ~ 244	94 ~ 544
Peizer (soft)	1,026		
Peizer (hard)	113		

The computing time of the Peizer approximation is problem independent. Therefore this approximation is prefered when a > 932 and the exact value of H(a) is not necessary. The hardware approach to the recursive exact method is much faster than other exact methods in all cases. It is faster than Peizer approximation using single CPU when a < 932.

5.5 Summary and Conclusions

In this chapter, we reviewed some existing techniques for computing the c.d.f. of the hypergeometric distribution. Since the direct computation is very time consuming, an approximation is desired in many cases. The recursive algorithm for the exact computation avoids much of the repetition of the direct method. We proposed a hardware pipeline architecture which implements the recursive formula and presented a detailed design. The main part of the design is actually a possible solution for a more general problem. This pipeline architecture is computationally efficient. For a large range of values of parameter a, the proposed architecture is 100 times to 10,000 times faster than the direct computation, 50 times to 100 times faster than the recursive method using single CPU, and two to ten times faster than the Peizer approximation using a single

CPU. This design uses six functional units for pipelines with no more than 15 segments. The modularity and the regularity of the system make it suited for VLSI implementation.

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH

6.1 Conclusions

The statistic S has a known distribution under a permutation null hypothesis and its critical level and threshold are easy to determine. This statistic is a good similarity measure between features in preliminary analysis for binary features. It has about the same power as the correlation coefficient. The statistic S is a good similarity measure between feature and category in binary tree classifier design. It gives similar trees as the mutual information criterion. The statistic S is also a reasonable adequacy measure for binary partitions in cluster validity. It is different but highly correlated to G_{amma} and other commonly used measures. In image template matching, S acts as well as other sub-optimal similarity measures between sub-images. An approximatation to the likelihood statistic is proposed. The hardware architecture

design is more efficient than other methods for computing hypergeometric c.d.f.'s.

The original contributions of this thesis are summarized below:

(1) Adaption of the cumulative hypergeometric distribution to the definition of similarity measures in pattern recognition.

(2) Definition of the preliminary feature analysis problem in pattern recognition and the use of S in this analysis (Chapter 2).

(3) Study of the relation among adequacy measures for binary partitions and the successful application of S in binary tree classifier design (Chapter 3).

(4) Definition of alternative hypothesis in image template matching and the design of a modified Neyman-Pearson statistic (Chapter 4).

(5) Design of a hardware implementation for computing the hypergeometric c.d.f. which is much faster than conventional means (Chapter 5).

6.2 Future Reseach

Future research includes extension to multivalued vectors, extension to two sided tests in preliminary feature analysis, extension to multi-class tree classifier design, and the multiple stage approach to image template matching.

6.2.1 Extension to Multi-Valued Vectors

This thesis involved binary vectors. The definition of a permutation statistic for multivalued nominal vectors is a direct extension, if we can define "match" between nominal values. When the match is obvious from common sense or from the physical meaning of the variables, the definition of S is straightforward. An example is given in Appendix E for a special case. Since not very many similarity measures exist for general nominal vectors, S could be an alternative. It will have known distribution as in the binary case. Since the computational complexity is high even for a moderate number of values, some algorithm or hardware structure must be used to apply this multi-valued permutation statistic to the problems in this thesis.

6.2.2 Two-Sided Tests for Preliminary Analysis

In Chapter 2, the tests are one sided since we are only interested in positive relations between features. This is true for questionnaire data, but not true for the cases where the coding (0,1) is irrelevent. Our statistic could be max(|S(i,j)-0.5|) and still have known distribution. It will be interesting to compare it with max(|C(i,j)|).

6.2.3 Multi-class Tree Classifier Design

Tree classifiers have been used for classification problems with a large number of pattern classes. At each node, the recognition problem is to identify which subset to which the unknown pattern belongs. If the tree itself is a binary tree, each node still deals with binary partitioning of a pattern set. A mulitivalued version of S' or some hierarchy of the binary version of S' might be used to assess the association of features to the grouping of class symbols.

6.2.4 Multi-stage and Sequential Approach to Template Matching

In image template matching a multi-stage approach might have certain advantages over the two-stage approach. An analysis of optimal number of stages in a hypothesis testing framework could be a future topic. It makes no sense to scan the entire image after the true location of the object has been found. A sequential decision making scheme under our hypothesis testing might be more efficient than the traditional approach. The hardware implementation of the computation of the proposed statistics may also be a future research topic.

APPENDIX A

THE THRESHOLDS AT THE BOUNDARIES OF RUNS GIVE HIGHEST S VALUES

Let $h(n_1,i)$ denote $h(n,n_1,n_2,i)$ for fixed n,n_2, where $h(.,.,.,.)$ is defined as in Sec. 2.1; and let

$$H(n_1,n_{11}) = \sum_{i=0}^{n_{11}} h(n_1,i)$$

We need to prove that $H(n_1,n_{11}) < H(n_1+1,n_{11}+1)$ for $n_1+n_2-n \leq n_{11} \leq \min(n_1,n_2)$. We proceed by induction on n_1.

Base case ($n_1 = 1$):

(1) If $n_{11}=0$,

$H(1,0)=(n-n_2)/(2n)$

$H(2,1)=(n-n_2)(n+n_2-1)/[n(n-1)] > (n-n_2)/(2n)$

(2) If $n_{11}=1$,

$H(1,1)=(n+n_2)/(2n)$

$H(2,2)=1 > (n+n_2)/(2n)$

Assume $H(n_1,n_{11}) < H(n_1+1,n_{11}+1)$ for a particular n_{11}.

(1) If $n_{11}' \leq \min(n_1, n_2)$

Since $$H(n_1+1,i)=\frac{(n_1+1)(n-n_1-n_2+i)}{(n-n_1)(n_1+1-i)}H(n_1,i)$$

$$H(n_1+2,i)=\frac{(n_1+2)(n-n_1-n_2+i)}{(n-n_1-1)(n_1+1-i)}H(n_1+1,i+1)$$

$$0< c_i=\frac{(n_1+1)(n-n_1-n_2+i)}{(n-n_1)(n_1+1-i)} < d_i=\frac{(n_1+2)(n-n_1-n_2+i)}{(n-n_1-1)(n_1+1-i)}$$

Therefore,

$$\sum_{--}^{--}[H(n_1,i)-H(n_1+1,i+1)] < \sum_{--}^{--}[c_iH(n_1,i)-d_iH(n_1+1,i+1)] < 0$$

(2) If $n_2>n_1$ and $n_{11}'=n_1+1$,

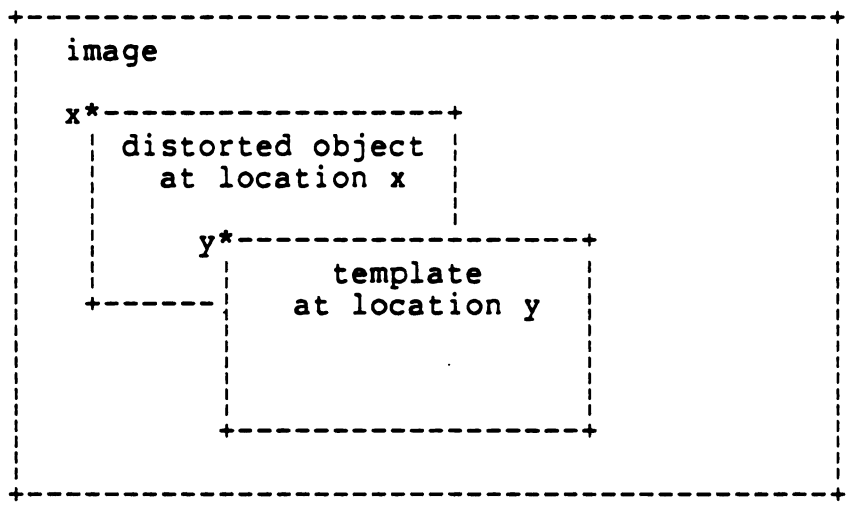
$$H(n_1+2,n_1+2)=\frac{(n_1+2)(n_2-n_1-1)}{(n_1+2)(n-n_1-1)}H(n_1+1,n_1+1)<H(n_1+1,n_1+1)$$

APPENDIX B

APPROXIMATION OF S_{xy} WHEN $y \neq x$

This section shows the derivation of S_{xy} (Sec. 4.2.1).

The following figure demonstrates the situation when a distorted object is at location x and a template is at location y where L_x overlaps with L_y.



$$S_{xy} = \frac{Pr(\ E_y \mid H_{1x})}{Pr(\ E_y \mid H_0)}$$

$$= \frac{Pr(\ N_{00}(y)= n_0 \mid H_{1x})\ Pr(\ N_{11}(y)=n_1 \mid H_{1x})}{Pr(\ N_{00}(y)= n_0 \mid H_0)\ Pr(\ N_{11}(y)=n_1 \mid H_0)}$$

Let t_{xy} be the overlap between L_y and L_x expressed as a number between 0 and 1. If x=y, t_{xy}=1, which results in S_{xx}, defined in Sec.4.2.1. If t_{xy}=0, S_{xy}= 1. For $0<t_{xy}<1$, the number of (0-0) matches can be written as the sum of two random variables

$$N_{00}(y) = N_{00}'(y) + N_{00}''(y).$$

where $N_{00}'(y)$ counts matches in the region of overlap and $N_{00}''(y)$ counts matches between the template and background. Under our assumptions, $N_{00}'(y)$ has a $B[t_{xy}n(1-q),1-p'']$ distribution and $N_{00}''(y)$ has a $B[(1-t_{xy})n(1-q),1-p]$ distribution, where

$$p'' = a(1-p')+bp'$$

and p' is the probability that a template pixel is 1. The distribution of the sum can be approximated by a Poisson distribution which involves three table look-ups and factorial computations. We choose to approximate this distribution by a $B[n(1-q),w_{xy}]$ distribution, where

$$w_{xy}= (1-p)(1-t_{xy})+(1-p'')t_{xy}$$

From Sec.4.2.1, $N_{00}(y)$ has a B[n(1-q),1-a] distribution under H_0.

A similar analysis shows that the distribution of $N_{11}(y)$ under H_{1x} can be approximated by a $B[nq,1-w_{xy}]$ distribution and $N_{11}(y)$ has a $B[nq,p]$ distribution under H_0.

Under these conditions,

$$S_{xy} \simeq \left[\frac{w_{xy}}{(1-w_{xy})}\,\frac{p}{(1-p)}\right]^{n_0(y)-n_1(y)} C_{xy}$$

$$\text{where } C_{xy} = \left[\frac{1-w_{xy}}{p}\right]^{n(1-q)} \left[\frac{w_{xy}}{1-p}\right]^{nq}$$

APPENDIX C

RESULTS OF t-TESTS FOR SEC. 4.3

We use PWD, PWJ, PWC, PWS and PWG to denote powers of D_x, J_x, C_x, S_x and G_x respectively. In the following table, each box reports the result of testing the hypothesis that Power(i)>Power(j) where i (row) and j (column) can refer to any of the five statistics above. The first number of each entry is the actual t value and the second number is w_t where the critical level is w_t. Positive t value indicates the acceptance of the hypothesis with level w_t; a negative t value indicates the acceptance of the reverse hypothesis. For example, Power(D_x)>Power(J_x) is accepted at level 0.36; Power(C_x) < Power(G_x) is accepted at level 0.10.

	PWJ	PWC	PWS	PWG
PWD	.367 (.36)	.472 (.32)	.501 (.31)	-0.748 (.23)
PWJ		.092 (.46)	.122 (.45)	-1.146 (.13)
PWC			.031 (.49)	-1.289 (.10)
PWS				-1.321 (.09)

This table indicates that G_x is more powerful than all others and D_x is slightly more powerful than the other three.

APPENDIX D

SUMMARY OF RESULTS FOR SEC. 4.3.1

The following table compares analytical approximations of the powers of D and G. In each box, the first integer is the number of combinations of (p,p",a,b) for which Power(D) < Power(G); the second number is the number of cases for which Power(D) > Power(G). There are 18 cases for each box.

image size / template size

subtemplate size	16x16 / 8x8	32x32 / 12x12
19	10 , 4	10 , 4
26	12 , 2	12 , 2
33	15 , 2	8 , 4
39	11 , 5	11 , 6

We consider all 144 pairs of power approximations as random samples and perform a standard two sample t-test. The resulting t value is 2.16648 and the hypothesis Power(D) < Power(G) is accepted at significence level (1-0.975).

APPENDIX E

AN EXAMPLE OF THE PERMUTATION STATISTIC IN MULTIVALUED NOMINAL VECTOR CASES

Consider nominal vectors V_1 with 3 possible values and V_2 with 4 possible values. We defined "match" between the values of vector components as follows. Let n be the vector length. The letter "x" means the corresponding values in the two vetors are considered as "match".

		V_2			
		1	2	3	4
V_1	1			x	x
	2		x		
	3	x	x		

Let n be the vector length, n_{i*} be the number of components taking value i in V_1, n_{*j} be the number of components taking value j in V_2, and let x_{ij} be the number of components that V_1 takes value i and V_2 takes value j. The following table indicates the relation among them.

		V_2				
		1	2	3	4	
	1			x_{13}	x_{14}	n_{1*}
V_1	2		x_{22}			n_{2*}
	3	x_{31}	x_{32}			n_{3*}
		n_{*1}	n_{*2}	n_{*3}	n_{*4}	n

Let the random experiment is the permutation of V_2, a random variable M be the number of matches between V_1 and V_2, and let m be the observed value of M in the original vector pair, then a permutation statistic S can be defined as

$$S = Pr(M \leq m) - Pr(M=m)U$$

where U is as defined in Chapter 1 and

$$\binom{n}{n_{1*}\ n_{2*}\ n_{3*}}^{-1} \sum_{x_{ij}} \binom{n_{*1}}{x_{11}\ x_{21}\ x_{31}} \cdots\cdots \binom{n_{*4}}{x_{14}\ x_{24}\ x_{34}}$$

LIST OF REFERENCES

LIST OF REFERENCES

1. M.R.Anderberg, Cluster Analysis for Applications, Academic Press, New York, 1973.

2. P.Argentiero, R.Chin and P.Beaudet, "An automated approach to the design of decision tree classifiers", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.PAMI-4, pp.51-57, Jan. 1982.

3. F.B.Baker and L.J.Hubert, "Measuring the power of hierarchical cluster analysis," J. of American Statistical Association, Vol.70, pp.31-38, Mar.1975.

4. R.R.Bahadur, "A representation of the joint distribution of response to n dichotomous items". Studies in Item Analysis and Prediction, H.Soloman (Ed.), Palo Alto, Cali.: Stanford Univ. Press, pp.158-168, 1961.

5. D.I.Barnea and H.E.Silverman, "A class of algorithms for fast digital image registration," IEEE Trans. Computers, Vol.C-21, No.2, pp.179-186, 1972.

6. P.L.Brockett, P.D.Haaland and A.Levine, "Information theoretic analysis of questionnaire data". IEEE Trans. Information Theory, Vol.IT-27, pp.438-445, 1981

7. T.M.Conover, Practical Nonparametric Statistics,

John Wiley & Sons, 2nd ed., New York, 1980.

8. T.M.Cover, "The two best independent measurements are not the best two". IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-4, pp.116-117,1974.

9. T.M.Cover and J.M.VanCampenhout, "On the possible orderings in the measurement selection problem", IEEE Trans. System, Man and Cybernetics, Vol. SMC-7, pp.657-661, 1977.

10. Cray research, Cray-1 Computer System Hardware Reference Manual 2240004, 1977.

11. P.J.Diggle, "On parameter estimate and goodness-of-fit testing for spatial point patterns," Biometrics, Vol.35, pp.87-101, March, 1979.

12. R.Dubes and A.Jain, "Clustering techniques: the user's dilemma," Pattern Recognition, Vol.8, pp.247-260, 1976.

13. R.Dubes and A.Jain, "Validity studies in clustering methodologies," Pattern Recognition, Vol.11, pp.235-254, 1979.

14. J.D.Elashoff, R.M.Elashoff and G.E.Goldman, "On the choice of variabless in classification problems with dichotomous variables", Biometrika, Vol.54, pp.668-670, 1967.

15. G.S.Fang, "A note on optimal selection of independent observations", IEEE Trans. Systems, Man and Cybernetics, Vol.SMC-6, pp.309-311, 1979.

16. R.V.Foutz, "A method for constructing exact tests from statistic that have unknown null distribution". Journal of Statistical Computation and Simulation, Vol.10. pp.187-193, 1980.

17. E.G.Fowlkes and C.L.Mallows,"A method for comparing two hierarchical clusterings," J. of American Statistical Association, Vol.78, pp.553-569, Sept.1983.

18. E.B.Fowlkes and C.L. Mallows, "Rejoinder", J. of American Statistical Association, pp.584, 1983.

19. J.H.Friedman and L.C. Rafsky, "Graph-theoretic measures of multivariate association and prediction", The Analysis of Statistics, Vol.11, No.2, pp.377-391, 1983.

20. K.S.Fu and T.Ichikawa, Special computer architecture for pattern processing. CRC Press, Boca Raton, Fla., 1982.

21. G.V.Glass and Hakstian,A.R., "Measures of association in Comparative experiments: their development and interpretation". American Educational Research Journal, pp.403-414, 1969.

22. D.W.Goodall, "A probabilistic similarity index", Nature, 1964.

23. D.W.Goodall, "A new similarity index based on probability", Biometrics, pp.882-907, 1966.

24. D.W.Goodall, "Numerical taxonomy of bacteria - some published data re-examined", Journal of General Microbiology, Vol.42, pp.25-37, 1966.

25. L.A.Goodman and Kruskal,W.H., "Measures of association for cross classifications". J. of American Statistical Association, pp.732-764, 1954.

26. L.A.Goodman and Kruskal,W.H., "Measures of association for cross classifications. II: further discussion and references", J. of American Statistical Association, Vol.54, pp.123-163, 1959.

27. L.A.Goodman and Kruskal,W.H., "Measures of association for cross classifications. III: approximate sample theory", J. of American Statistical Association, Vol.58, pp.310-364, 1963.

28. L.A.Goodman and Kruskal,W.H., "Measures of association for cross classifications. IV: Simplification of asymptotic variances", J. of American Statistical Association, Vol.67, pp.415-447, 1972.

29. A.Goshtasby, S.H.Gage, and J.F.Bartholic, "A two-stage cross correlation approach to template matching," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.PAMI-6, No.3, pp.374-378, May 1984.

30. Z.Hubalek, "Coefficients of association and similarity, based on binary (presence, absence) data - an evaluation", Biological Review, Vol.57, pp.669-689, 1982.

31. L.Hubert and J.Schultz, "Quadratic assignment as a general data analysis strategy", British Journal Math. Statist. Psychol., Vol.29, pp.190-241, 1976.

32. L.Hubert and J.R.Levin, "A general statistical framework for assessing categorical clustering in free recall", Psychological Bulletin, Vol.83, No.6, pp.1072-1080, 1976.

33. L.Hubert, "The comparison and fitting of given classification schemes", Journal of Mathematical Psychology, Vol.16, No.3, pp.233-253, 1977.

34. L.Hubert, "Generalized proximity function comparisons", British Journal Math. Statist. Psychol., Vol.31, pp.179-192, 1978.

35. L.Hubert, "Generalized concordance", Psychometrika, Vol.44, No.2, pp.135-142, 1979.

36. L.Hubert and Subkoviak,M.J., "Confirmatory Inference and Geometric Models", Psychological Bulletin, Vol 86,No.2,361-370, 1979.

37. L.Hubert and R.G.Golledge, "A heuristic method for the comparison related structures", Journal of mathematical psychology, Vol.23, pp.214-226, 1981.

38. G.E.Hughes,"On the mean accuracy of statistical pattern recognizers", IEEE Trans. Information Theory, Vol.IT-15, pp.420-421, Jan. 1969.

39. K.Hwang, Computer Arithmetic, John Wiley & Sons, New York, 1979.

40. A.K.Jain and R.C. Dubes, "Feature definition in pattern recognition with small sample size", Pattern recognition, Vol.10, pp.85-97, 1978.

41. M.F.Janowitz, "Similarity measures on binary data", Systematic Zoology, pp.342-359, 1980.

42. I.T.Jolliffe and B.J.T. Morgan, "Comment", J. of American Statistical Association, pp.580-581, 1983.

43. M.D.Kelly, "Edge detection in pictures by computer using planning," Matching Intelligence 6 (D. Michie,Ed), pp.379-409, Edinburgh Univ. Press, Edinburgh, 1971.

44. O.Kempthorne, Design and analysis of experiments , John Wiley & Sons, N.Y., 1952.

45. J.B.Kruskal, "Multidimensional scaling and other methods for discovering structure," Statistical Methods for Digital Computers, pp.296-339, 1977.

46. A.Kulkarni and L.Kanal, "An optimization approach to hierarchical classifier design", 3rd

International. Joint Conf. on Pattern Recognition, pp.459-466, 1976.

47. A.Kulkarni, "On the mean accuracy of hierarchical classifiers", IEEE Trans. Computers, Vol.C-27, No.8, pp.771-776, Aug. 1978.

48. A.Kulkarni, "Admissible search strategaies for paramtric and nonparametric hierarchical classifiers", 4th International. Joint Conf. on Pattern Recognition, pp.238-248, 1978.

49. M.Kurzynski, "The optimal strategy of tree classifier", Pattern Recognition, Vol.16, No.1, pp.81-87, 1983.

50. G.Landeweerd, T.Timmers, E.Gelsema, M.Bins and M.Halie, "Binary tree versus single level tree classification of white blood cells", Pattern Recognition, Vol.16, No.6, pp.571-577, 1983.

51. Y.K.Lin and K.S.Fu, "Automatic classification of cervical cells using a binary tree classifier", Pattern Recognition, Vol.16, No.1, pp.69-80, 1983.

52. R.F.Ling and J.W.Pratt, "The accuracy of Peizer approximations to the hypergeometric distribution, with comparisons to some other approximations," J. of American Statistical Association, Vol.79, No.385, pp.49-60, Mar.1984

53. N.Mantel, "The detection of disease clustering and a generalized regression aproach," Cancer Research, Vol.27, pp.209-220, Feb.1967.

54. N.Mantel and R.S.Valand, "A technique of nonparametric multivariate analysis", Biometrics, Vol.26, pp.547-558, Sept.1970.

55. W.Meisel and D.Michalopoulos, "A partitioning

algorithm with application and the optimization of decision trees", IEEE Trans. Computers, Vol.C-22, No.1, Jan.1973.

56. G.W.Milligan, "A Monte Carlo study of thirty internal criterion measure for cluster analysis," Psychometrika, Vol.46, pp.187-199, June 1981.

57. H.Moravec and D.Gennery, Cart Project Progress Report, Stanford Artif. Intell. Project, Internal Memo. Oct. 1976.

58. A.N.Mucciardi and E.E.Gose, "A comparison of seven techniques for choosing subsets of pattern recognition properties", IEEE Trans. Comput., Vol.20, pp.1023-1031, 1971.

59. J.Mui and K.S.Fu, "Automated classification of nucleated blood cells using a binary tree classifier", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.PAMI-2, No.5, pp.429-443, Sept.1980.

60. C.Munteanu, "Evaluation of the sequential similarity detection algorithm applied to binary images," Pattern Recognition, Vol.13, No.2, pp.167-175, 1981.

61. R.N.Nagel and A.Rosenfeld, "Ordered search techniques in template matching," Proc. IEEE, Vol.60, pp.242-244, 1972.

62. J.W.V.Ness,"Comment", J. of American Statistical Association, pp.576-579, 1983.

63. L.M.Ni and A.K.Jain, "A VLSI systolic architecture for pattern clustering," to appear on IEEE Trans. Pattern Analysis and Machine Intelligence.

64. L.M.Ni and K.Hwang, "Vector reduction techniques

for arithmetic pipelines," to appear on IEEE Trans. on Computers.

65. H.Payne and W.Meisel, "An algorithm for constructing optimal binary decision trees", IEEE Trans. Computers, Vol.C-26, No.9, pp.905-916, Sept.1977.

66. H.K.Ramapriyan, "A multilevel approach to sequential detection of pictorial features", IEEE Trans. Computers, Vol.C-25, No.1, pp.66-78, 1976.

67. W.M.Rand, "Objective criteria for evaluation of clustering methods," J. of American Statistical Association, Vol.66, pp.846-850, 1971.

68. A.Rosenfeld, Picture processing by computer. New York, Academic Press, 1969.

69. A.Rosenfeld and G.J.Vanderbrug, "Coarse-fine template matching", IEEE Trans. System, Man and Cybernetics, Vol.SMC-7, No.2, pp.104-107, 1977.

70. E.Rounds, "A combined nonparametric approach to feature selection and binary decision tree design", Pattern Recognition, Vol.12, pp.313-317, 1980.

71. C.E.Sarndal, "A comparative study of association measures", Psychometrika, Vol.39, pp.165-187, 1974.

72. J.Schuermann and W.Doster, "A decision theoretic approach to hierarchical classifier design", Pattern Recognition, Vol.17, No.3, pp.359-369, 1984.

73. I.Sethi and B.Chatterjee, "Machine recognition of constrained hand printed Devanagari", Pattern Recognition, Vol.9, pp.69-75, 1977.

74. I.Sethi and G.Sarvarayudu, "hierarchical classifier design using mutual information", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.PAMI-4, No.4, pp.441-445, July 1982.

75. Q.Y.Shi and K.S.Fu, "A method for the design of binary tree classifiers", Pattern Recognition, Vol.16, No.6, pp.593-603, 1983.

76. J.Slagle and R.Lee, "Application of game tree searching techniques to sequential pattern recognition", Communication of ACM, Vol.14, No.2, Feb.1971.

77. P.H.A.Sneath and R.R.Sokal, Numerical taxonomy, W.H.Freeman. SanFrancisco. 1973.

78. Special Issue on feature extraction and selection in pattern recognition, IEEE Trans. Comput. Vol.20, 1971.

79. W.Stallings, "Approaches to Chinese character recognition," Pattern Recognition, Vol.8, pp.87-98, 1976.

80. M.Svedlew, C.D.McGillem and P.E.Anuta, "Experimental examination of similarity measures and preprocessing methods used for image registration," Symp. Machine Processing of Remotely Sensed Data, 1976.

81. P.Swain and H.Hauska, "The decision tree classifier: design and potential", IEEE Trans. Geoscience Electronics, Vol.GE-15, No.3, pp.142-147, July 1977.

82. S.L.Tanimoto, "Template matching in pyramids," Computer Graphics and Image Processing, Vol.16, pp.356-369, 1981.

83. D.J.Vanderbrug and A.Rosenfeld, "Two-stage template matching," IEEE Trans. Computers, Vol.C-26, no.4, pp.384-393, 1977.

84. D.L.Wallace, "Comment", J. of American Statistical Association, pp.569-567, 1983.

85. Q.R.Wang and C.Y.Suen, "Analyis and design of a decision tree based on entropy reduction and its application to large character set recognition", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.PAMI-6, No.4, pp.406-417, 1984.

86. A.Whitney, "A direct method of nonparametric measurement selection", IEEE Trans. Computers. Vol.C-20, pp.1100-1103, 1971.

87. R.Y.Wong and E.L.Hall, "Scene matching with invariant moments," Computer Graphics and Image Processing, Vol.8, pp.16-24, 1978.

88. R.Y.Wong and E.L.Hall, "Sequential hierachical scene matching," IEEE Trans. Computers, Vol.c-27, No.4, April 1978.

89. R.Y.Wong and E.L.Hall, "Performance comparison of scene matching techniques," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.PAMI-1, No.3, July 1979.

90. K.C.You and K.S.Fu, "An approach to the design of a linear binary tree classifier", 1976 Machine Processing of Remotely Sensed Data, pp.3A-1 - 3A-10.