5-1-1987

Hybrid pattern recognition

Eustache Placide

Atlanta University

Follow this and additional works at: http://digitalcommons.auctr.edu/dissertations

Part of the Mathematics Commons

Recommended Citation

http://digitalcommons.auctr.edu/dissertations/3018

This Thesis is brought to you for free and open access by DigitalCommons@Robert W. Woodruff Library, Atlanta University Center. It has been accepted for inclusion in ETD Collection for AUC Robert W. Woodruff Library by an authorized editor of DigitalCommons@Robert W. Woodruff Library, Atlanta University Center. For more information, please contact cwiseman@auctr.edu.
HYBRID PATTERN RECOGNITION

A THESIS
SUBMITTED TO THE FACULTY OF ATLANTA UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE

BY
EUSTACHE PLACIDE

DEPARTMENT OF MATHEMATICAL SCIENCE

ATLANTA, GEORGIA

MAY, 1987
ABSTRACT

COMPUTER SCIENCE

PLACIDE, EUSTACHE

B.S. MORRIS BROWN COLLEGE, 1985

HYBRID PATTERN RECOGNITION

Advisor: Dr. Nazi A. Warsi

Thesis dated May, 1987

There are two basic approaches to pattern recognition: decision-theoretic and syntactic. However, in actual applications, a combination of both may be needed. One such hybrid technique consists of syntactic method coupled with stochasticity in its grammar. Randomness in the syntactic case is caused due to noise and insufficient information about characteristics of pattern classes. To absorb the effect of this randomness, the grammar must be generalized to include the probabilities of production rules.

In this paper, a preliminary discussion of issues involved with hybrid techniques, in general, and stochastic grammars, in particular, is provided. An efficient algorithm for an automatic learning of production probabilities is devised. Concepts are illustrated via examples.
ACKNOWLEDGMENTS

I would like to convey my special thanks and appreciation to Dr. Nazir A. Warsi, my advisor, who has been my mentor and has given me his valuable time and help in writing this thesis. I am thankful to Drs. Benjamin J. Martin and Bennett Setzer who helped me get started in this program. I would also like to thank Professor Steve Ornburn and other faculty members who have supported and encouraged me. Finally, I would like to thank Sandra Rucker, Enock, Evy, Louisa and Charles Hubert Placide who have helped me in my endeavors. This research was partially supported by N.A.S.A. Langley Research Center. (Grant Number: NAG-1-412).
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>ii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iii</td>
</tr>
<tr>
<td><strong>Chapter</strong></td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. BASIC APPROACHES TO PATTERN RECOGNITION</td>
<td>4</td>
</tr>
<tr>
<td>Decision-Theoretic and Syntactic Approaches</td>
<td></td>
</tr>
<tr>
<td>Hybrid Approach</td>
<td></td>
</tr>
<tr>
<td>III. STOCHASTIC GRAMMARS AND LANGUAGES</td>
<td>13</td>
</tr>
<tr>
<td>Stochastic Grammars</td>
<td></td>
</tr>
<tr>
<td>Stochastic Languages</td>
<td></td>
</tr>
<tr>
<td>IV. AUTOMATIC LEARNING OF PRODUCTION</td>
<td>21</td>
</tr>
<tr>
<td>PROBABILITIES</td>
<td></td>
</tr>
<tr>
<td>Estimation of Production</td>
<td></td>
</tr>
<tr>
<td>Probabilities</td>
<td></td>
</tr>
<tr>
<td>An Algorithm for Learning Production</td>
<td></td>
</tr>
<tr>
<td>Probabilities</td>
<td></td>
</tr>
<tr>
<td>An Example</td>
<td></td>
</tr>
<tr>
<td>V. CONCLUSION</td>
<td>23</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>31</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

Pattern recognition is useful in many applications. Its techniques are among the most important tools used in the field of machine intelligence. Pattern recognition can be defined as a categorization of input data into identifiable classes via extraction of significant features or attributes of the data from a background of irrelevant details. [1, 2].

The two methods generally used in the solution of pattern recognition problems are based on decision-theoretic and syntactic approaches. The decision-theoretic approach is best suited for applications in which patterns can be represented in vector forms. On the other hand, the syntactic approach is used when the structure of a pattern is significant to the classification process. The syntactic approach uses the similarities between the structure of patterns and the syntax of a language in its classification process. The interest in the use of stochastic languages for syntactic pattern recognition arose in part because of serious drawbacks in the decision-theoretic approach. Drawbacks stem from the fact that the decision-theoretic approach is not designed to handle
pattern structures and their relationships. For such cases, the syntactic approach must be employed.

For an automatic pattern recognition system using the syntactic concept, the main issues are: formulation of problem, pattern description, creation of an automatic recognizer/parser, and absorption of stochastic variations in the recognition process. This paper deals with an introduction of the syntactic approach. In a random environment, stochastic techniques must be utilized to describe and characterize variables. Two major factors contribute to this randomness: noise due to measurement, and insufficient information about pattern class characteristics. This paper deals with the generalization of the formal grammar models to include the stochastic framework. An efficient algorithm for an automatic learning of production probabilities is constructed. An up-to-date detailed bibliography is provided to aid further research efforts on this topic.
REFERENCES


CHAPTER II

BASIC APPROACHES TO PATTERN RECOGNITION

Decision-Theoretic and Syntactic Approaches

Pattern recognition problems may be grouped into two general approaches: the decision-theoretic approach and the syntactic approach. In the decision-theoretic approach, the features are extracted from the patterns and the recognition of each pattern is usually made by partitioning the feature space. Many research efforts dealing with the decision-theoretic approach and its applications exist in literature. Different learning techniques and decision rules have been applied to estimate the pattern characteristics of various classes. This approach has been successfully used in solving practical problems such as character recognition, medical diagnosis, etc. [3-13].

The structure information that describes each pattern is important in some pattern recognition problems. In such cases, the recognition process embraces not only the capability of assigning the pattern to a particular class, but also the potentiality to describe aspects of the pattern that make it ineligible for assignment to another class. One example of this class of recognition problem is
scene analysis. The decision-theoretic approach has not been very efficient for this problem. The decision making is based on statistical decision rules. The origin of concepts used in syntactic pattern recognition can be traced to the formal language theory. Initially the aim was to develop mathematical models that could describe natural languages. If this had been accomplished, the next step would be to program computers to interpret natural languages for problem solving and translation purposes. The afore-mentioned aim has not been realized thus far. However, an extension of this type of research has significantly influenced the field of pattern recognition.

The purpose of a pattern description language is to provide the structural description of patterns in terms of a set of pattern primitives and their composition operations. We may think of primitives as being symbols permissible in some grammar, where a grammar is a set of rules of syntax for the generation of sentences from the given symbols [14]. One of the basic steps in the syntactic pattern recognition method is the decomposition of patterns into subpatterns or primitives. Figure 1 illustrates the decomposition of two structures into relations of the primitives. By following each structure in a clockwise direction, it is certainly possible to detect and encode these primitives in the form of a string of qualifiers. Therefore, the triangle can be represented
by the string acb, and the rectangle by the string aaacceeedd [15-27].

Fig. 1. Triangle and Rectangle
Furthermore, it is possible to envision two grammars Gs1 and Gs2 whose rules allow the generation of sentences that correspond to the triangle and the rectangle. Therefore, the language $L(Gs1)$ generated by Gs1 consists of sentences representing the triangle and $L(Gs2)$ generated by Gs2 of sentences representing the rectangle [28-38]. It is first necessary to establish a grammar which consists of the rules for the generation of sentences in a language. Once the grammar corresponding to a pattern class has been defined, we can then proceed with the syntactic pattern recognition process. Input patterns can be represented by sentences. The problem then consists of deciding in which language the sentence corresponding to the input is valid. Thus, if the sentence belongs to $L(Gs1)$, we say it is in the class whose grammar is Gs1. If the sentence is invalid over both $L(Gs1)$ and $L(Gs2)$, the input pattern is assigned to a rejection class consisting of invalid patterns [39-47].

Inherent in the nonstochastic syntactic pattern recognition is the assumption that patterns being considered are equally likely to occur. Practically speaking, there is a need to formulate an approach which considers cases where some pattern variations are more likely than others. Probabilities can then be used in the classification process to improve the quality of recognition.
Hybrid Approach

Syntactic approach to pattern recognition provides capability for describing a large set of complex structures by using small sets of simple pattern primitives and their grammatical relations. In the syntactic approach the pattern is represented as strings, trees, and graphs with pattern primitives and relations. The decision-making process consists of parsing.

In syntactic approach, effects of randomness may appear at two places: primitives and recognition. The effect of noise at the primitive level involves smoothing algorithms using dynamic programming optimization techniques. However, randomness at the recognition level may be caused due to an addition factor: lack of complete information about pattern class characteristics. To absorb the effect of the noise at this level, stochasticity must be built into the grammar by assigning a probability to each production. This directly depends on the probability of a production being used in derivation of strings being recognized. This calls for using a hybrid approach in which the usual grammars are generalized by incorporating stochastic elements.
REFERENCES


CHAPTER III

STOCHASTIC GRAMMARS AND LANGUAGES

Stochastic Grammars

A stochastic grammar is a five tuple $G_s = (N, \Sigma, D, Q, R)$. Elements that constitute $G_s$ can be defined in the following manner:

$N$ is a finite set of nonterminals or variables;
$\Sigma$ is a finite set of terminals or constants;
$D$ is a finite set of productions or rewriting rules;
$Q$ in $N$ is the starting symbol;
$R$ is the set of probabilities assigned to the productions of $D$ [48-57].

The stochastic grammar is specified by the assignment of probabilities to the productions of $D$. Consider the following derivation of a terminal string $Y$ which starts at $Q$, i.e.:

$Q \rightarrow b_0 \rightarrow b_1 \rightarrow b_2 \ldots \rightarrow b_m \rightarrow Y$

We can view this derivation as a sequence of stochastic events. Then the probability of generating $b_{i+1}$ from $b_i$ is the probability of applying the production labelled $s_{i+1}$, which is denoted by $d(s_{i+1})$. Continuing in this manner, the probability of the generation of $Y$ using
The production sequence $S^n_1 S^n_2 \ldots S^n_i$ is given by

$$D(Y) = d(s_i) d(s_i/s_{i-1} s_{i-2}) \ldots d(s_1/s_{i-1} s_{i-2} \ldots s_2).$$

Here $<3(S^n_1 S^n_2 \ldots s_{i-1})$ is the conditional probability of applying the rule $S_i$ having applied $S_{i-1}, S_{i-2}, \ldots S_2, S_1$.

A stochastic grammar that consists of productions permitting more than one derivation of a terminal string is called ambiguous.

Example:

The sentence $a + b^* c$ has two leftmost derivations:

$$a + b^* c \quad a + b^* c \quad a + b^* c \quad a + b^* c \quad a + b^* c \quad a + b^* c \quad a + b^* c.$$
In some sense the parse tree of figure (a) is correct because it reflects the commonly assumed precedence of + and * while figure (b) does not. Hence, if there are \( n \) distinct leftmost derivations of a terminal string \( Y \) with respective probabilities \( d_1(y), d_2(y), \ldots, d_n(y), n > 1 \), then the probability of generation of \( Y \) is \( d(y) = \sum_{i=1}^{n} d_i(y) \), which is the sum of the probabilities of a finite number of mutually exclusive events.

Stochastic grammars are used in syntactic pattern recognition problems with finite number of pattern classes, where each class is defined by a distinct pattern grammar but with possible random errors [71]. Given cases where a string to be classified may be an element of more than one language, it is not evident which classification is most appropriate. Stochastic decision theory can be used to assign these strings into pattern classes if the grammars and sentences in them can be assigned probabilities which reflect their likelihood. For a given string \( Y \), the value is computed for each pattern grammar \( G_1, G_2, \ldots, G_n \), \( 1 \leq i \leq n \). The maximum value of \( D(G_i/Y) \) determines which pattern class \( Y \) is assigned to. We note that \( D(G_i) \) is a priority probability defined in the following manner:

\[
D(G_i/Y) = \frac{D(y/G_i)D(G_i)}{D(y)}
\]
Stochastic Languages

The stochastic language $L(G_s)$ generated by a stochastic grammar $G_s$ is the set of derivable terminal strings together with their probabilities of generation:

$$L(G_s) = \left\{ [y, d(y)]/y \text{ in } \Sigma^*, Q \overset{*}{\rightarrow} y, i = 1, 2, \ldots, n, \right.$$

$$d(y) = \sum_{i=1}^{n} D_i(y),$$

where $n$ is the number of distinct leftmost derivations of $y$. The fact that there is a specific derivation with probability $D_i(y)$ is indicated by writing $Q \overset{D_i(y)}{\rightarrow} Y$. If the grammar is unambiguous, there is a simpler definition: $L(G_s) = [y, d(y)] : y \text{ in } Q \overset{d(y)}{\rightarrow} Y$.

Two stochastic grammars $G_{s1}$ and $G_{s2}$ are equivalent if $L(G_{s1}) = L(G_{s2})$ [72-74]. To illustrate the above concepts, let us consider the following example:

Consider the following grammar $G_s$ given by

$$N = \{ Q \},$$

$$\Sigma = \{ a, b, c \},$$

Production:

1. $Q \rightarrow abQC$, probability of (1) = $D$
2. $Q \rightarrow c$, probability of (2) = $1-D$

This grammar generates strings of type $Y = (ab)^i c^{i+1}$ where rule (1) is applied $i$ times followed one application of rule (2). Thus, the probability of generation of $Y$ is given by $D^i (1-D)$. Hence $L(G_s) = \left\{ [(ab)^i c^{i+1}, D^i (1-D)] : i \geq 1 \right\}$. We also note that $\sum_{i=1}^{\infty} D^i (1-D) = (1-D) \ast 1 : (1-D) = 1$. In such situations, the probability set of productions is called consistent.

Languages that are used to describe the noisy and distorted patterns are often ambiguous in the sense that
one string can be generated by more than one grammar which specifies the patterns generated from a particular pattern class.
REFERENCES


CHAPTER IV

AUTOMATIC LEARNING OF PRODUCTION PROBABILITIES

Estimation of Production Probabilities

Assume that \( x_h : 1 \leq h \leq n \) is a sample set of patterns with frequencies \( f(x_h) \) of \( x_h \) in training samples. Given a grammar \( G_s \), the probability of \( x_h \) being in \( L(G_s) \) is \( P(G_s / x_h) \). Therefore, the relative frequency of \( x_h \) in \( L(G_s) \) is given by \( f(x_h) \times P(G_s / x_h) \). With the above information, we will determine on an average how many times a grammar is used in recognizing all sample patterns. With this in mind, we can parse the pattern \( x_h \) and count how many times a rule is used in one derivation of \( x_h \). An average number of times a grammar rule is used in all derivations of \( x_h \) is given by the product of the count with the relative frequency of \( x_h \). The sum for sample patterns determines an average total number of times a rule is used.

To find the production probability, we divide rules into groups such that all rules with the same left side are in one group. With this format, the cumulative sum of average totals in each group can be determined. Now in order to find the probability for a production, we take a group, and divide the total average for each production by the
cumulative total average [75-36]. This leads to the following Algorithm.

**An Algorithm For Learning Production Probabilities**

**Input:**
1. $\{G^t : 1 \leq t \leq q\}$ set of grammars with
   
   $G^t = \{x^t, N^t, Q^t, S^t\}$,
   
   $Q^t = \{A^t_i \rightarrow B^t_{ij} : 0 \leq i \leq m^t, 1 \leq j \leq l^t_i, A^t_0 = S^t\}$

2. Sample Patterns $P = \{x_h : 1 \leq h \leq m\}$ with $f(x_h)$ = frequency of $x_h$ in training samples, $P(G^t_s / x_h)$ = probability of $x_h$ being in $L(G^t_s)$.

**Algorithm:**
1. For $t = 1$ to $q$ do

2. For $h = 1$ to $n$ do

   Find $F^t_h = f(x_h)p(G^t_s/x_h)$;

   Parse $x_h$ using $G^t_s$ and find a derivation $S^t \rightarrow x_h$;

   For $i = 0$ to $m^t$ do

   For $j = 1$ to $l^t_i$ do

   Find $n^t_{ij}(x_h)$ = number of times $A^t_i \rightarrow B^t_{ij}$ used in $S^t \rightarrow x_h$.
Find \( E[A_i^t \rightarrow B_{ij}^t] = \sum_{h=1}^{m} n_{ij}^t(x_h)F_i^t \)

Find \( c^t[A_i^t] = \sum_{j=1}^{l_i} E[A_i^t \rightarrow B_{ij}^t] \)

For \( j = 1 \) to \( l_i^t \) do

Find \( p[A_i^t \rightarrow B_{ij}^t] = E[A_i^t \rightarrow B_{ij}^t]/c^t[A_i^t] \).

An Example

<table>
<thead>
<tr>
<th>Gs1</th>
<th>Gs2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) S</td>
<td>(1) S</td>
</tr>
<tr>
<td>A</td>
<td>aA</td>
</tr>
<tr>
<td>A</td>
<td>aA</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>bB</td>
</tr>
<tr>
<td>(4) B</td>
<td>(5) B</td>
</tr>
<tr>
<td></td>
<td>b</td>
</tr>
</tbody>
</table>

Suppose that class \( w_1 \) consists of strings of a's, and class \( w_2 \) of strings of b's; however, strings may sometimes occur because of noise corruption. Notice that Gs1 and Gs2 can both produce mixed strings. Let \( P = x_1, x_2, x_3 \) where \( x_1 = aaab, x_2 = aabb, \) and \( x_3 = ab \). Let frequencies of occurrences of these strings in sample set be given by

\[
\begin{align*}
f(x_1) &= 30, \
f(x_2) &= 20, \
f(x_3) &= 10.
\end{align*}
\]

Probabilities of strings to be recognized by each grammar is given by

\[
\begin{align*}
p(Gs_1/x_1) &= 1, & p(Gs_1/x_2) &= .5, & p(Gs_1/x_3) &= 0, \
p(Gs_2/x_1) &= 0, & p(Gs_2/x_2) &= .5, & p(Gs_2/x_3) &= 1.
\end{align*}
\]
$F^n_1$ and $F^n_2$ values are given the following tabular form.

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>$F^n_1$</th>
<th>$F^n_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

Derivations: $Gs_1$

$S \xrightarrow{(1)} aA \xrightarrow{(2)} aaA \xrightarrow{(3)} aaaA \xrightarrow{(5)} aabB \xrightarrow{(5)} aab = x_1$

$S \xrightarrow{(1)} aA \xrightarrow{(2)} aaA \xrightarrow{(4)} aabB \xrightarrow{(5)} aabB = x_2$

$S \xrightarrow{(1)} aA \xrightarrow{(3)} aB \xrightarrow{(5)} aB = x_3$

Derivations: $Gs_2$

$S \xrightarrow{(1)} aA \xrightarrow{(2)} aaA \xrightarrow{(3)} aaaA \xrightarrow{(5)} aabB \xrightarrow{(5)} aab = x_1$

$S \xrightarrow{(1)} aA \xrightarrow{(3)} aaB \xrightarrow{(4)} aabB \xrightarrow{(5)} aabB = x_2$

$x_3$ cannot be parsed.
Calculations for $G_{s_1}$:

<table>
<thead>
<tr>
<th>$x_h$</th>
<th>$n_{11}$</th>
<th>$n_{21}$</th>
<th>$n_{22}$</th>
<th>$n_{31}$</th>
<th>$n_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_h$</th>
<th>$n_{11}^F$</th>
<th>$n_{21}^F$</th>
<th>$n_{22}^F$</th>
<th>$n_{31}^F$</th>
<th>$n_{32}^F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>30</td>
<td>60</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>$x_2$</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sum</td>
<td>40</td>
<td>70</td>
<td>30</td>
<td>10</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tot</th>
<th>Tot</th>
<th>Tot</th>
</tr>
</thead>
<tbody>
<tr>
<td>40/40</td>
<td>70/100</td>
<td>30/100</td>
</tr>
</tbody>
</table>

$P[S \rightarrow aA] = 1$  
$P[A \rightarrow aA] = 7/10$  
$P[A \rightarrow B] = 3/10$  
$P[B \rightarrow bB] = 1/5$  
$P[B \rightarrow b] = 4/5$
Calculations for $G_2$:  

<table>
<thead>
<tr>
<th>$x_h$</th>
<th>$n_{11}$</th>
<th>$n_{21}$</th>
<th>$n_{22}$</th>
<th>$n_{31}$</th>
<th>$n_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_h$</th>
<th>$n_{11}F_n^2$</th>
<th>$n_{21}F_n^2$</th>
<th>$n_{22}F_n^2$</th>
<th>$n_{31}F_n^2$</th>
<th>$n_{32}F_n^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sum</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tot = 10</th>
<th>Tot = 10</th>
<th>Tot = 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/10</td>
<td>0/10</td>
<td>10/20</td>
</tr>
</tbody>
</table>

$P[S \rightarrow aA] = 1$  
$P[A \rightarrow aA] = 0$  
$P[A \rightarrow aB] = 1$  
$P[A \rightarrow bB] = 1/2$  
$P[A \rightarrow b] = 1/2$
REFERENCES


CHAPTER V
CONCLUSION

Pattern recognition is useful in many applications such as biomedical data analysis, diagnostic decision making, target detection and identification, failure analysis, and diagnosis of engineering system.

Many results have been obtained in various applications of syntactic approaches to problems such as the character recognition and chromosomal analysis. However, the dichotomy of syntactic and decision-theoretic approaches appears to be convenient only from the viewpoint of theoretical studies. Many pre-processing techniques are useful in both approaches, apparently because the selection of a particular pre-processing technique depends more on the type of the pattern under study than the recognition approach being applied.

The feature extraction and selection problem in the decision-theoretic approach and the primitive extraction and selection problem in the syntactic approach are similar in nature except that primitives in the syntactic approach represent subpattern and, on the other hand, features in the decision-theoretic approach may be any set of numerical measurements taken from the pattern.
Stochastic languages have been proposed for the description of noisy or distorted patterns. Another application of stochastic languages is in the learning of grammars from actual pattern samples. The properties of stochastic grammars are important for their applications, both in theory and in applications.

This paper deals with an introduction to principal ideas encountered in the syntactic concept of pattern recognition in the presence of randomness. Although the primitive selection problem considered here concerns string grammars, the concept can be extended to multidimensional grammars. However, due to variations in juxtapositioning of multidimensional structures, the selection of such grammars is difficult. Stochastic considerations can be incorporated into the recognition process via production probabilities. The main problem is to learn such probabilities automatically from training samples. Although the concept is extendable to multi-dimensional grammars, it becomes more complicated due to juxtapositional variations of structures.

For an efficient production probability learning system, the speed of the employed parsing system is crucial. Once probabilities have been learned, one can build a stochastic recognizer utilizing the probability of each production. For example, if a handle in the bottom-up parsing is to be reduced at a certain stage and there are several prospective productions with this handle on their
right hand sides, the one with the highest production probability is chosen. It is evident from the algorithm presented earlier that this grammar has the largest probability of having been applied in generation of this handle.

For an efficient recognition, it is also important to build an automatic grammar learning system. Although rudimentary efforts exist on this topic, a lot is left to be derived. Many of the currently known techniques are heuristic in nature, which is especially true in multidimensional cases.
BIBLIOGRAPHY


Ahmed, P., and Suen, C. Y. Edge Classification and Extraction of Shape Features. Montreal, Quebec, Canada: Concordia University, 1970.


