A Comparative Study of Pattern Recognition and Artificial Intelligence Techniques For the Development of Intelligent Systems

Sheila A. McIlraith

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Research Report CS-85-11
May 1985

A Comparative Study of Pattern Recognition and Artificial Intelligence Techniques for the Development of Intelligent Systems

Sheila A. McIlraith

An essay
presented to the University of Waterloo
in partial fulfillment of the
requirements for the degree of
Masters of Mathematics
in
Computer Science

Supervisor: Marlene Jones
Waterloo, Ontario, April 1985

Abstract

Over the past three decades, Artificial Intelligence (AI) has been transformed from a field concerned with modelling human cognition on a computer, to a computing discipline intent on the construction of programs that perform difficult human tasks. With this transformation has come the utilization of many computing techniques, originally deemed contrary to the theme of human cognition. Pattern Recognition (PR) has developed tools that address some of the problems being faced by AI researchers. The purpose of this paper is to evaluate and compare Pattern Recognition and Artificial Intelligence techniques for developing intelligent systems, and to indicate areas where PR techniques could be incorporated into an AI framework. The paper has been written from an Artificial Intelligence perspective. It is assumed that the reader has the equivalent background to an introductory course in AI. No prior knowledge of Pattern Recognition is required.

An intuitive introduction to Pattern Recognition principles is provided. It contains a summary of pertinent techniques in both decision-theoretic and syntactic PR. PR and AI techniques are compared with respect to methodology, formalization, ease of implementation, ease of understanding and modification, domain applicability, and potential for future expansion. Three specific areas of AI are isolated for more in depth study: knowledge representation, problem-solving techniques and learning. Within each area, a comparison of PR and AI is provided, and suggestions are made for the application of PR techniques to the AI environment.

Acknowledgements

I would like to thank my supervisor, Dr. Marlene Jones, who allowed me to work on this topic even though it was not directly related to her own research interests. She gave me a lot of helpful suggestions and encouragement. She was also the catalyst for much of the interaction between graduate students and faculty both in courses and at special interest group meetings. I would also like to thank Dr. Larry Rendell of the University of Illinois for constructive comments and encouragement, and Dr. Mohamed Kamel of the University of Waterloo for his interest in this material. Finally, I would like to recognize the friends I have made here at the University of Waterloo for their support and for all the good times we shared. The academic and social interaction among students and faculty helped inspire this work.

Sheila McIlraith

Table of Contents

A Comparative Study of Pattern Recognition and Artificial Intelligence Techniques for the Development of Intelligent Systems

1. Introduction

In the last three decades, a great deal of research has been done in the area of machine intelligence, with the ultimate goal of creating an *intelligent* computer system. This label has, perhaps inaccurately, been applied to a set of problems whose common features include the fact that they can be performed by humans; they cannot always be solved by conventional Von Neumann programming techniques; and they require a great deal of knowledge. Medical diagnosis, chemical analysis, game playing, theorem proving, speech understanding and image analysis are some examples of machine intelligence problems.

Computers are becoming an integral part of society. There is a desire to build computer systems that are user-friendly so that the general public may have access to them and enjoy their use. Bridging the gap between man and machine requires giving the machine a semblance of human intelligence; natural language understanding, game playing and intelligent computer assisted instruction (ICAI) assist to this end.

The incorporation of computers into day-to-day life has produced an information explosion. Suddenly, researchers in many diversified fields have the ability to store and to process large quantities of data. This has escalated the production of research results which in turn generates more information. "Experts", such as doctors, are then faced with the problem of retaining and utilizing these results. Decision-making is becoming more and more difficult to perform accurately. The application of intelligent computer systems to emulate these experts' reasoning processes, given a vast amount of knowledge, is a logical solution to the problem.

Solely as a research topic, machine intelligence is fascinating. It requires the design and implementation of a unique style of computing. Computer researchers have approached this problem in a variety of ways. Some have concentrated on hardware, trying to produce machines capable of performing parallel computations analogous to the parallelism in the brain. Others have invested their time in the development of software techniques such as **Pattern Recognition** (PR) and **Artificial**

Intelligence (AI), to perform human-like tasks. It is these techniques that are the subject of this paper.

PR and AI began as one. One of the original domains for machine intelligence research was scene analysis [25]. The problem was to develop a generalized procedure for recognizing objects in a detailed scene. These objects were to be recognized, regardless of their size, orientation or detail.

Two different techniques were applied to the problem. PR researchers attempted to use multivariate statistical methods for representing and recognizing objects or *patterns*. These methods had no correlation to human techniques for performing the same task, but they worked. Certainly that was enough to warrant development of what is now a very active and successful methodology for problems of machine intelligence.

Another, very different approach was taken to the same problem of scene analysis. AI researchers attempted to draw on research done by psychologists into how humans analyze scenes. In doing so, they tried to develop a computer model of human cognition and then to structure their computing accordingly. From AI came the concept of symbolic computing and with it, new programming languages, new methods of knowledge representation and perhaps to come new styles of computing machines. AI techniques were not as immediately successful as PR techniques. However, a whole new style of computing was created based on the manipulation of symbols This has yielded many successful applications and many extensions of the original computing methodology.

There is an analogy to be drawn between the techniques of PR and AI, and the principles behind **Engineering** and **Science**. *Engineering* is

> *"the application of scientific principles to practical ends as in the design, construction and operation of efficient and economical structures, equipment and systems."* [24].

There is a direct concern with creating a finished product that is efficient and economical. Scientific principles are applied, but to "practical ends". This is parallel to the methodology behind PR. Statistical techniques have been applied to problems in machine intelligence to create an efficient and economical system. No emphasis has been placed on trying to capture the methodology of the human mind.

Conversely, AI more closely parallels scientific methodology. *Science* has been defined as

> *"Systematic and formulated knowledge; Branch of knowledge (esp. one that can be conducted on scientific principles), organized body of the knowledge that can be accumulated on the subjet"* [33].

Additionally, the word *scientific* has been defined as

> *"according to rules laid down in exact science for performing observations and testing soundness of conclusions, systematic, accurate; (of act or agent) assisted by expert knowledge"*. [33]

Systematic and formulated knowledge is the foundation for all AI systems. Knowledge is accumulated by an expert in the problem domain and stored in an efficient, yet meaningful manner. There is an attempt to retain a direct mapping between the functioning of the computer system and the real world model. As a result, AI as a study of knowledge and intelligent behaviour is also of interest to researchers in a broad spectrum of fields including psychology, philosophy, linguistics, education and epistemology.

To describe AI as a field to develop computer models of human cognition is inaccurate. AI has used human cognition as a model to guide its development. However, there are radical physical limitations to what a computer can do. AI researchers are computer scientists and as such, one of their mandates is to create "efficient and economical" systems, regardless of the technique. Similarly, PR has expanded from its original multivariate statistical methods to include more semantic processing. More emphasis is being placed on retaining a mapping to the real world model. The two techniques are less polar than when originally conceived.

PR and AI do have different approaches to machine intelligence. Consequently, each method seems to be better suited to a subset of the existing problems. PR techniques are more successful at giving computers sensory capabilities. Character recognition, speech recognition and computer vision have all been successfully tackled. These problems are often characterized by analog input that must be transformed before the mathematical reasoning process may be implemented. On the other hand, AI techniques seem to be better suited to performing human thinking tasks. Medical diagnosis, natural language understanding, and intelligent computer assisted instruction are just a few of the domains to which AI principles have been applied. Although these problem areas can be seen as being different, there is still some overlap in the applicability of both techniques.

It is important for researchers to keep abreast of research in other areas of computer science and engineering. VLSI design methodology has been modelled after the design of large operating systems (OS), using the idea of virtual machines in OS to abstract the complex design problem. Data communication has used many of the results from graph theory and statistics to assist in modelling and designing efficient networks. Syntactic PR, to be described later, has used formal language theory. There are many more examples that could be cited. PR and AI are fertile fields for similar cross-pollenation.

To date, there has been little interaction between the fields of PR and AI. In general, PR and AI researchers have been critical of the other's approach to machine intelligence problems. Perhaps professional rivalry has played a part in the lack of communication and shared effort between the two groups. There is a need for increased cooperation. The academic community has been slowly accepting this fact. In 1976 the first IEEE Joint Workshop on Pattern Recognition and Artificial Intelligence was held. Selected topics in PR and AI, of interest to both groups were discussed. In 1979 the IEEE Transactions on Pattern Analysis and Machine Intelligence Journal was created to assist in communication between AI researchers and PR researchers.

Despite this effort at shared literature, researchers have been slow to attempt to incorporate their counterpart's techniques. PR researchers seem to have been more active in this area than AI researchers. A few of the existing PR systems have started to use AI techniques for high level semantic processing. AI techniques are much better suited for representing and dealing with meta level knowledge than PR. There are many areas of AI where PR techniques could be used to supplement existing methods. There is a need for more migration of theory form PR to AI.

The human mind uses statistical uncertainty even though the human may not explicitly recognize it as such. Many of the decisions made in a human's daily life involve weighing and evaluating choices based on some certainty factors. Humans allow a degree of variance in criteria when making decisions. Therefore it does not seem contrary to the methodology of AI purists to include some statistical or probabilistic computations in the decision-making process. MYCIN, an expert system that addresses the problem of diagnosing and treating infectious blood diseases, is one of the few AI systems to implement certainty factors or numeric plausibility factors into its reasoning architecture. PR has many decision-making algorithms based on statistical and probabilistic theory that could assist in the AI decision-making process.

The purpose of this paper is to evaluate and compare PR and AI techniques for developing intelligent systems, and to indicate areas where PR techniques could be incorporated into an AI framework. The original purpose of this paper was to cite these applications in detail. Further reading and work displayed the enormity of this task. PR is a vast field of study, generating volumes of literature annually. To fully represent the field and to compare it to AI would require several texts. The goal of this paper, therefore, is to educate and to provoke creative thought. It is hoped that this paper will give the AI reader a strong introduction to the field of Pattern Recognition while relating it to an Artificial Intelligence framework. There are many areas presented only briefly in this paper where much work could be done. It is recommended that the reader interested in applying Pattern Recognition techniques to Artificial Intelligence, select a specific subarea upon which to work. Hopefully, this will assist in opening the channels of communication between AI researchers and PR researchers and stimulate some combined work in the two areas. The paper has been written from an Artificial Intelligence perspective. It is assumed that the reader has the equivalent background to an introductory course in AI. No prior knowledge of Pattern Recognition is required. A supplementary reading list of material for further technical details in both areas is provided. The author has attempted to present the material objectively, while trying to encourage the reader to consider PR as a tool for AI applications.

Chapter 2 contains an introduction to Pattern Recognition principles. An intuitive description of the area is given, with little technical detail included. The interested reader is referred to the supplementary readings. Chapter 3 contains a comparison of PR and AI techniques with respect to certain metrics. Chapters 4, 5 and 6 deal with knowledge representation, problem solving techniques and knowledge acquisition respectively; current PR and AI techniques in each of these subcategories are compared. Proposals for the use of PR principles in an AI framework are given. Chapter 7 is a concluding chapter. It summarizes important points made herein and compiles suggestions for future research.

2. Introduction to Pattern Recognition

What **is** *Pattern Recognition?* A reader having seen and understood that sequence of symbols has just experienced pattern recognition first hand although he/she may not have known it. Somehow those inanimate black patterns on white paper were sensed, analysed and interpreted. The reader somehow gained some meaning from them. What is of interest to scientists is how this was done and how it can be emulated by a computer. Certainly the functioning of the eye is understood. The printed characters are focussed on the retina where they are sensed and transformed into signals to be interpreted by the brain. What are those signals? How does the brain discern between an "r" and an "n"? How does it so easily recognize those letters when they are hand written or typed in different fonts? Ironically, humans are experts at performing this task but they can't explain how they do it. Regardless of this lack of full understanding, automatic character recognition systems have been created for restricted input. Character recognition is just one application of pattern recognition techniques.

In human beings, Pattern Recognition (PR) is a perceptual process. Input is either sensory or conceptual. Sensory input includes such physical stimulus categories as vision, hearing, smell, touch and taste. Conceptual input is more abstract. It includes such patterns as solution strategies and argument approaches. This input is sensed, analysed and recognized. Perhaps it is a stimulus that has been sensed before or perhaps it is similar to something of which the human has previous knowledge. The human then assimilates this input with existing knowledge to classify it or to improve on his/her knowledge store.

In a machine, PR is a data analysis technique emulating a perceptual process. Its purpose is to give a machine perceptual capabilities. The machine accepts measurements representative of a certain object, analyses them, and then classifies the object as being most *similar* to a model or prototype of a known object stored in memory. Formally, Pattern Recognition can be defined as

> *"The categorization of input data into identifiable classes via the extraction of significant features or attributes of the data from a background of irrelevant detail."*[34]

There is a popular misconception that PR techniques are only suitable for problems such as computer vision, remote sensing and speech recognition. Problems that are characterized by input with little immediate meaning to a human, input such as analog data. This has been supported by the fact that most of the PR applications are in areas related to those mentioned above. It must be stressed that PR provides general techniques for the categorization of input data into identifiable classes. These techniques are applicable to any problem domain where decision making or classification based on data is required. Perhaps a failure to understand this feature of PR techniques has been a cause for its lack of use by AI researchers.

There are two types of PR, **Decision-theoretic** PR and **Syntactic** PR. Decision-theoretic PR is the classical Pattern Recognition approach, based on mathematical principles. It commonly uses probabilistic and multivariate statistical techniques to represent and classify patterns. Syntactic or Linguistic PR is slightly closer to the AI approach to machine intelligence. It is frequently used for analyzing and recognizing patterns that are not easily represented by numerical measurements alone. The syntactic approach attempts to represent a pattern as a hierarchy of subpatterns. These subpatterns are just primitive patterns with relations to connect them. The hierarchy of subpatterns is analogous to the hierarchy of a parse tree or language derivation. Consequently, many of the theories of formal language have been applied to these pattern grammars.

2.1. The Generic Pattern Recognition System

The creation of a generic Pattern Recognition system consists of two stages, the analysis stage, and the recognition stage [18]. In general, the analysis stage involves taking sets of samples, representative of a finite number of classes, and training the PR system to distinguish between them. This requires selecting representative features from each set of samples and defining a suitable classifier that will efficiently and accurately be able to classify unknowns into one of the classification groups.

If the samples used to represent the class are sufficient, then the analysis stage is complete. In most applications of machine intelligence problems, it is difficult to fully represent a class by a small number of samples. Therefore, some automatic PR systems contain an adaptive or learning element. An approximate classifier is implemented, based on the original samples. This classifier is then iteratively refined by the learning element procedure until a satisfactory level of classification is reached.

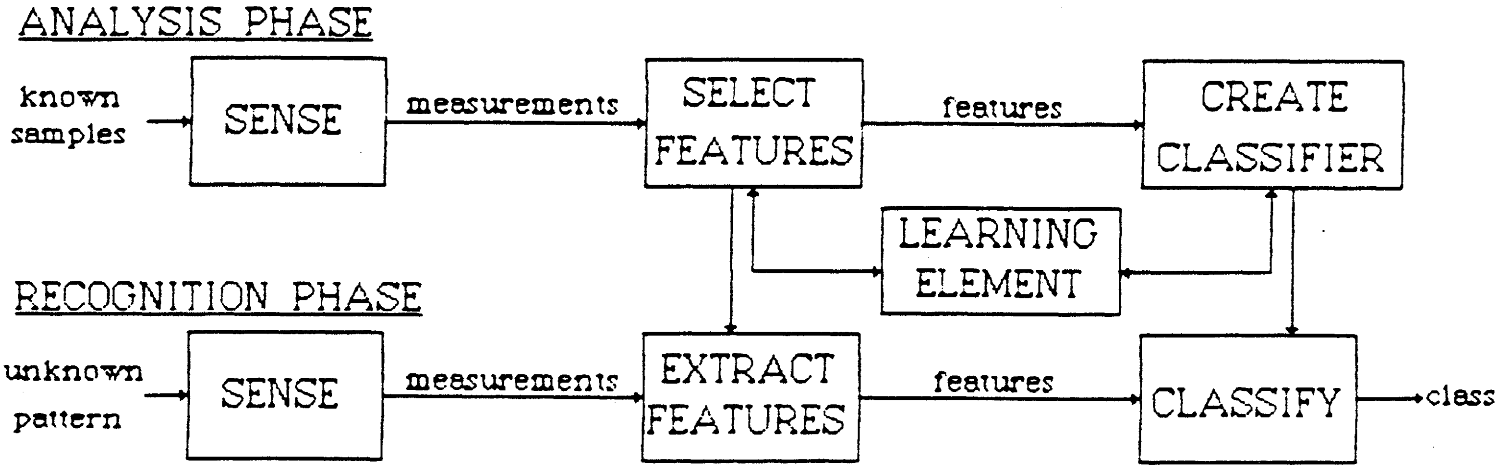


Fig 2.1 Generic Pattern Recognition System

Once the initial analysis phase of the PR process is complete, the recognition phase begins. This phase is comprised of several steps. The first step requires taking measurements from unknown patterns to be classified. These measurements are then transformed into the features selected in the analysis phase and required by the system's classifier. The pattern classifier is then applied to the extracted features to decide on a categorization or classification for the unknown input pattern. Figure 2.1 represents the operation of a generic Pattern Recognition system.

2.1.1. Specific Issues to be Resolved

Each step in the analysis and recognition phases of a PR system generates a unique set of problems. Initially, both the representative samples and any unknown patterns awaiting classification must be sensed. This requires taking measurements and expressing them in a form that the computer can use and understand. In many of the machine intelligence applications to which PR has been applied, this is not a trivial task. Often problems are a result of a lack of understanding of how humans perform the task, and therefore how to sense it to capture distinguishing attributes. For example, in a speech recognition system, a tape-recorder could be utilized to record an utterance, but this is not usable by the computer. A sound spectrogram is often employed to capture the characteristics of the analog signal. It is digitized and then further processed. Similarly, in image processing or computer vision problems, photographs are taken and then digitized to put them in computer usable form.

Knowledge representation (KR), although not labelled as such, is an issue in PR system design just as it is in AI system design. Once in computer usable form, features must be selected and extracted from sensed measurements. Features are a function of their measurements. One of the primary motivations for the translation of measurements into features is to use only independent and discerning attributes of a pattern in the classification process. By selecting features as a combination of measurements or as one of several redundant measurements, the dimensionality of the classification problem is reduced. The fewer features required to characterize a class, the simpler the classification algorithm.

Unfortunately, with the increase in speed and efficiency comes a loss of information. It is difficult to select features automatically to represent a class. Statistical techniques can be used to find linearly independent features with small within class variance and large between class variance. This does not guarantee a good characterization of the class, and it is not applicable to all problem domains. The feature selection and ultimate classification techniques applied are a function of the specific problem. The process is not yet totally automated. Often the intuition and experience of the system designer is required to make a PR application function satisfactorily. This will be discussed in further detail in subsequent sections.

The next step in the PR process, classification, is quite complex. Many standard classification techniques exist. The selection of a classification depends on the character of the data to be categorized. Two techniques frequently used for classification are *membership-roster* and *common property* [34]. The membership-roster technique involves representing a class as the list of all its individual members. Classification is performed by directly matching an unknown pattern with a particular member of the class. This method is not very sophisticated. It requires a great deal of memory to store all the members, and template matching is time consuming. However it is a good method for a small number of classes with little variability between samples of a class.

The common property approach is the conventional approach to classification. Features are selected to represent a class and various similarity measures are used to decide which class the features of an unknown resemble the most. When the features are representable numerically, statistical techniques can be used to measure similarity in feature space by representing classes as clusters of points in the n dimensions of the n features. When they are not numeric, different techniques must be used. The common property approach is much more flexible than the membership-roster approach. Feature matching allows for variation in the pattern to be classified. Similarity as opposed to equality is the criterion for classification. Another advantage is that less storage is required for the class representation. A disadvantage of the common property approach is that there is a loss of information in the translation of a class representation to a finite set of features. A solution would be to increase the number of representative features to include more information, at the expense of increased complexity of the resulting system.

The learning element of a PR system only operates during the analysis phase, or during a subsequent update to the system if performance is deemed unsatisfactory. Most learning elements in PR systems are involved in adaptive learning, trying to improve on initial classifiers from labelled sample input. There are applications where labelled sets of samples are not provided. In these cases, the system is only provided with a set of unlabelled training patterns. It must sort these unlabelled patterns into similar groups and then select discerning features of each group. This type of learning is often referred to as *learning without a teacher*.

Learning is an integral part of intelligent behaviour. It is a difficult problem in the area of machine intelligence. In Pattern Recognition systems, learning is displayed in several areas. The automatic selection of primitives or features from labelled samples, the inductive creation of a classifier from these features, the improvement of a classifier by an adaptive learning element, and finally the extraction of classes from a group of unlabelled samples, are all instances of machine learning.

2.2. An Illustrative Example

To better understand the Pattern Recognition process, a very simplistic example is given: the task of distinguishing between horses and donkeys. To most humans, this may seem straightforward. However, to define it in computer terms makes it a non-trivial problem. The sensing step of the analysis phase requires taking measurements from a herd of donkeys and a herd of horses. Many measurements may be acquired: height, number of legs, weight, body shape, colouring, ear size etc. A decision must be made as to whether the problem is better suited to a membership-roster or common property approach, and to decision-theoretic or syntactic PR techniques. Many of the measurements can be represented in numeric terms. Therefore, the common property, decision-theoretic approach will be selected. There is a limit statistically to the number of measurements that can be employed and still give reliable estimates of the mean, variance and perhaps distribution that may be used to represent a prototype for each class. Some measurements are of no use at all, others supply redundant information. For example, both donkeys and horses have 4 legs, thus this measurement gives no discriminating information. Similarly both donkeys and horses come in assorted combinations of colours. Caution must also be taken in adding features that are correlated. Take for example height and weight; these two measurements are definitely correlated within classes. Similarly, in distinguishing between races of people, skin type, eye colour and hair colour are also correlated. To avoid statistical degradation of the classifier, a linear combination of measurements may be used as a feature or all but one correlated measurement may simply be excluded. It is important to remember that the complexity of the classifier increases with the number of features. Thus a minimum number of features should be selected.

In the horse and donkey example, two features shall be selected: height and ear size. Thus every animal in the horse-donkey classification problem will be represented by a 2 dimensional vector, one dimension being height, the other ear size. The feature space upon which classification will occur is therefore two dimensional. If these features were plotted for a sample of horses and donkeys, a diagram similar to the one in Figure 2.2 might be seen.

Once the features have been selected, a classifier is chosen. There are many different ways of measuring similarity in a decision-theoretic approach. These will be expanded upon in the following section. The PR problem is then to classify an unknown, **u** of ear size u_1 and height u_2 by measuring its similarity to both classes using the selected classifier. The unknown pattern is assigned to the most similar class.

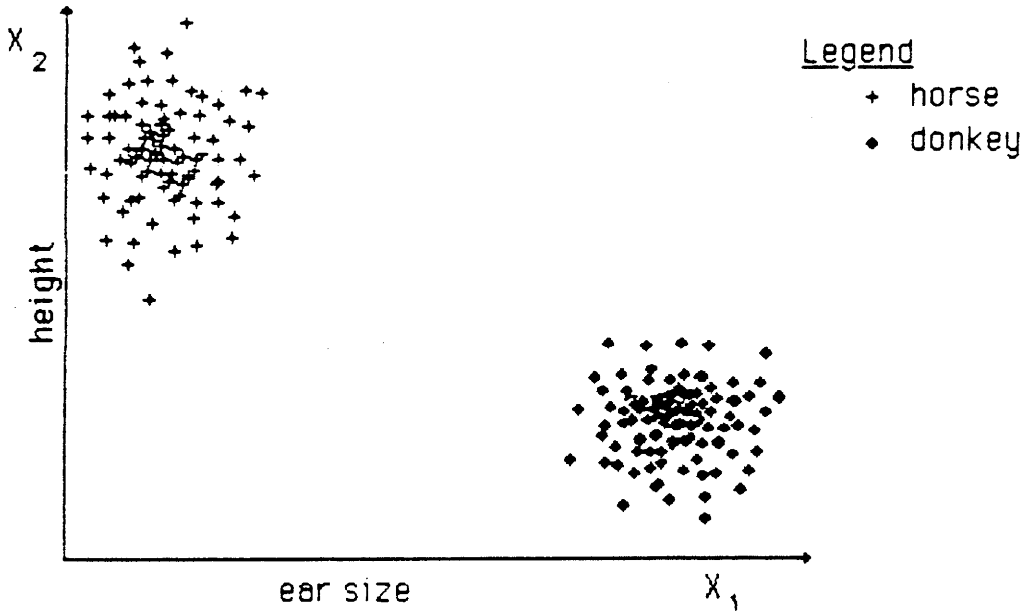


Fig 2.2 Horse and Donkey Training Samples Plotted in Feature Space

2.3. Specific Pattern Recognition Techniques

Pattern Recognition is a large and well established field of research. Each year, new techniques are proposed for improving sensing, feature selection/extraction, classification and learning. The following section gives the reader an intuitive feel for the basic methodology behind both decision-theoretic PR and syntactic PR. This should be sufficient background to understand comparisons performed in subsequent chapters.

2.3.1. Decision-Theoretic Pattern Recognition

2.3.1.1. Knowledge Representation

In decision-theoretic PR, an object/pattern may be represented by a vector of *m* measurements obtained from sensors. The task is to transform this measurement vector into a new *n*-dimensional vector of features ($n<m$). The features selected represent the defining attributes of the class. One of the main purposes of feature selection is to reduce the dimensionality of the pattern representation and thus reduce the complexity of the classification problem.

The horse-donkey example is an illustration of this pattern representation with $n=2$. They may be plotted in feature space. From the hypothetical graphing of sample horses and donkeys in Figure 2.2, it is easy to visualize the characteristics of good features. Classification could be performed quite accurately by drawing a diagonal line an equal distance between the two sets of samples. Unknown patterns located on one side of the diagonal line would be horses and on the other side, donkeys. The fact that the samples are clustered tightly together and that the two clusters are recognizably far apart, makes classification simple. Translated into more technical terms, ideal features would have a minimum *intra* or within class distance and a maximum *inter* or between class distance. Statistically, this is represented by features with little variance within a class and a great deal of variance between classes. This can be measured on input data and is one means by which features can be selected.

2.3.1.2. Classification

The real power in PR is its ability to take unknown samples and classify them into different groups/classes based on similarity. A class may be thought of as a group of patterns which are **similar** or equivalent. It may be represented by a prototype such as the mean, by a set of typical patterns belonging to that class, or by the statistical distribution the samples generate.

The beauty of the system is that patterns need not be identical to belong to the same class. A pattern feature may have a range of values because of normal variation or noise in measurements. By using statistical techniques, measures of similarity may be created to allow for these variations.

The act of classification involves assigning a class name or label to a pattern. There are three common types of classification problem. [18]

1. Each class is defined by a multivariate probability density function (pdf), $p(x|\text{class})$, which is either known or derived. Classification is performed by statistical decision-theory techniques to minimize the probability of error.
2. The pdf of each class is unknown. The class is represented by a set of labelled samples. Classification may be performed by statistically estimating the pdf and using statistical decision-theory techniques or by creating classification rules based on the distribution of the samples.
3. The only information given is a set of unlabelled samples. Clustering techniques are required to determine the number of classes and the definition of each class.

The problem in defining a classifier is in deciding on a good measure of similarity. Several possibilities for similarity measures will be discussed in the following sections. They may be broken down into several categories including distance metrics, metrics that make use of pdf information, trainable deterministic metrics and trainable statistical metrics.

Distance Classifiers

Suppose an unknown pattern is introduced into the horse-donkey classification problem. Its feature values are plotted in Figure 2.3. By visual inspection, the unknown should be classified as a donkey. Visual inspection may be formalized as a minimum distance measure between the unknown and the class. The class may be represented by a prototype value such as the mean of the labelled samples, or by some ideal form that best represents the class.

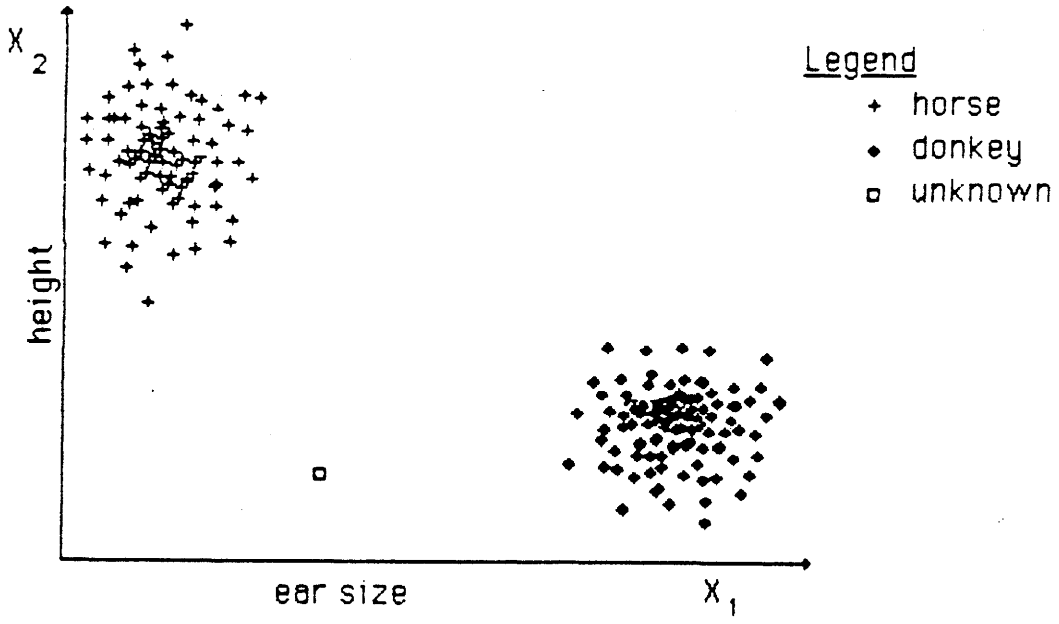


Fig 2.3 The Classification of an Unknown Pattern

a) Minimum Euclidean Distance (MED) Classifier

The simplest and least powerful of the distance metrics is minimum euclidean distance (MED). A prototype for each class is selected, for example the mean, and the euclidean distance is measured from the unknown to each prototype. The unknown is categorized as belonging to the class yielding the MED.

This metric is good because it is not sensitive to outliers or noise. However, it does not account for different variances for different features. For example, imagine that ponies were added to the class of horses. The variance in ear size might be marginally increased, but the variance in height of the horse class would increase dramatically. This increase in variance is illustrated in Figure 2.5 by the elliptical shape displayed by the horse class. The slight skew in the orientation of the ellipse with respect to the major axes represents a slight correlation between ear size and

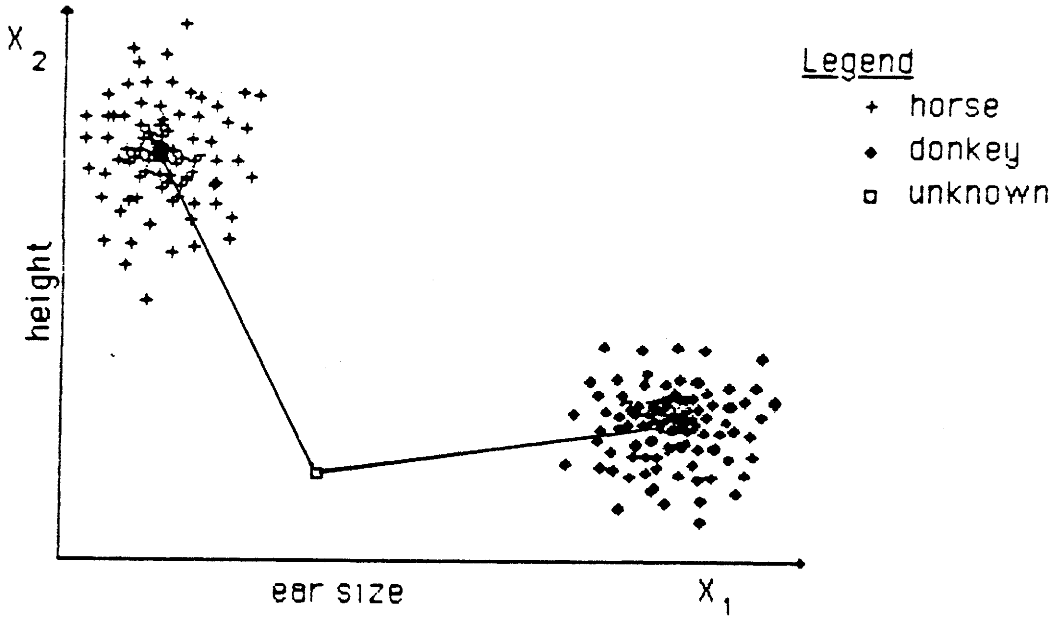


Fig 2.4 Minimum Euclidean Distance Metric

height. Smaller horses/ponies have slightly smaller ears.

Consider the unknown again. If the means are used as prototypes, the unknown would be classified as a donkey. However visual inspection indicates that the unknown is closer to the horse class if the variance is taken into account. Thus misclassification would occur. The MED classifier works best with uncorrelated, equal variance data.

An alternate application of the MED classifier would be to choose the **nearest neighbour** of the unknown as a prototype. To implement this, the distance between the unknown and all known samples of each class must be calculated. The class of the sample creating the MED is then selected. The problem with this system is that it is very sensitive to outliers and noise. Thus select the **K-nearest-neighbours** as the prototype [18]. Both prototypes have difficulties. Their implementation requires the individual storage of much of the labelled data. The calculations to find

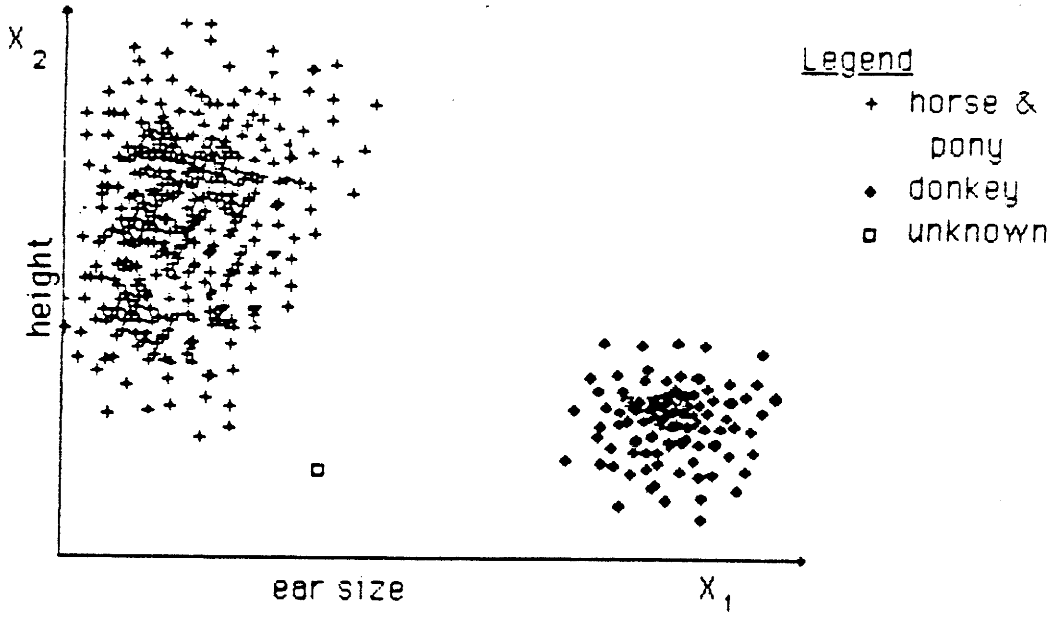


Fig 2.5 Inclusion of Ponies into the Horse Class

the prototype are numerous.

b) Minimum Intra-class Distance (MICD) Classifier

This distance classifier is based on the premise that within a class, samples should be as similar as possible. Figuratively speaking, classes should be represented by small diameter, tightly packed circles/ellipses. Thus, to classify an unknown sample using this logic, equidistance contours measured in standard deviation units are created around each class. These contours group possible patterns that are equally similar to the class. A decision boundary for classification can be created by connecting the intersection points of corresponding equidistance contours. Figure 2.6 illustrates this notion.

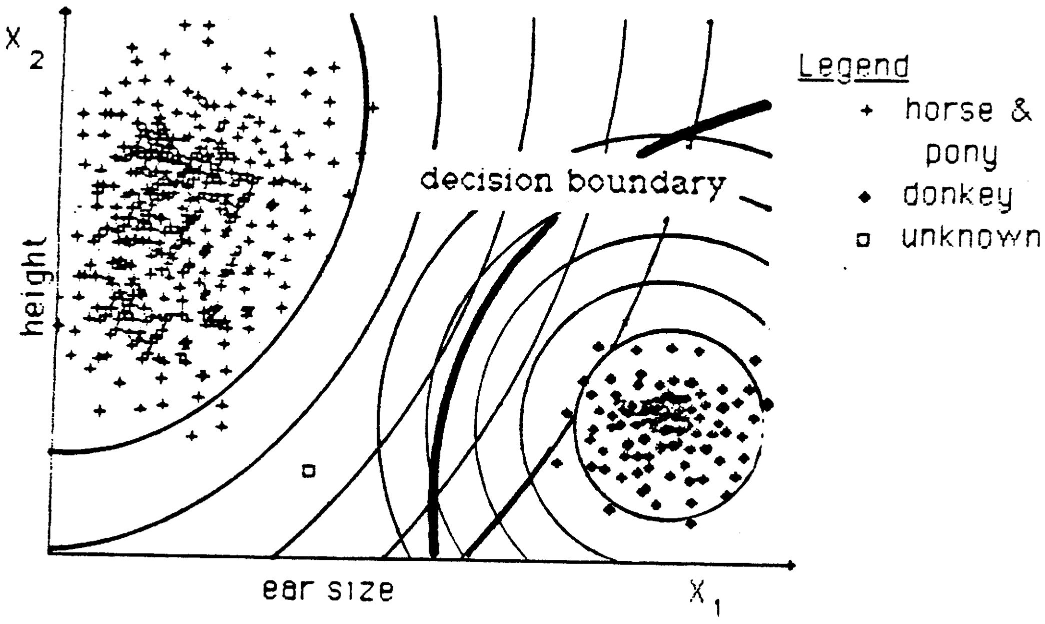


Fig 2.6 The Minimum Intra-class Distance Metric

The MICD classifier displays superior performance to the MED because it uses the mean and the variance in its class representation. MICD measures distance in standard deviations, thus it is unitless. The result is a decision surface in *n* space. It does not function very well for one class contained within another. Another problem with MICD is the question of whether mean and variance are sufficient descriptors of the class. A more powerful classifier yet is MAP.

Probabilistic Classifiers

Probabilistic Classifiers, in general, deliver better classification with a minimum probability of misclassification. A prerequisite for using probabilistic classifiers is knowledge of the probability density function (pdf) of the samples. Techniques exist to estimate unknown pdfs.

a) Maximum A Posteriori Probability (MAP) Classifier

If the pdf of a class is known, a superior MAP classifier may be used to classify patterns. Intuitively, an unknown pattern $\mathbf{x}$ is more likely to belong to the class which has greater probability given the value of $\mathbf{x}$. Reconsider the horse-donkey problem. For ease of illustration, the height feature will be removed. The only feature used for classification is the ear size. An unknown pattern of ear size x_1 is to be classified. Since there is only one feature, $p(\mathbf{x}) = p(x_1)$

$\mathbf{x}$ is a horse iff

$$P(\text{horse}|\mathbf{x}) > P(\text{donkey}|\mathbf{x})$$

However using Bayes theorem

$$P(\text{horse}|\mathbf{x}) = p(\mathbf{x}|\text{horse})P(\text{horse}) \,/\, p(\mathbf{x})$$

This may be transformed into:

$\mathbf{x}$ is a horse iff

$$p(\mathbf{x}|\text{horse})P(\text{horse}) > p(\mathbf{x}|\text{donkey})P(\text{donkey})$$

When the probability of either class occurring is equal then

$\mathbf{x}$ is a horse iff

$$p(\mathbf{x}|\text{horse}) > p(\mathbf{x}|\text{donkey})$$

This is illustrated in Figure 2.7.

When the class probabilities are not equal, the curves are weighted by the probabilities. Figure 2.8 illustrates the case when

$$P(\text{horse}) > P(\text{donkey}).$$

The classification threshold is shifted towards the mean of the class that is weighted less, in this case donkeys. This classifier uses a much more accurate description of the class. It also minimizes the probability of misclassification. The difficulty is that in most classification problems, the labelled data is given without knowledge of the underlying pdf. There are techniques for estimating the statistical nature of a class. When the form of the pdf is known, parameter estimation can be performed by maximum likelihood or Bayes estimation. When the form of the pdf is unknown the statistical nature of the class can be estimated by density estimation techniques such as parzen window or K-nearest-neighbour [12] [18].

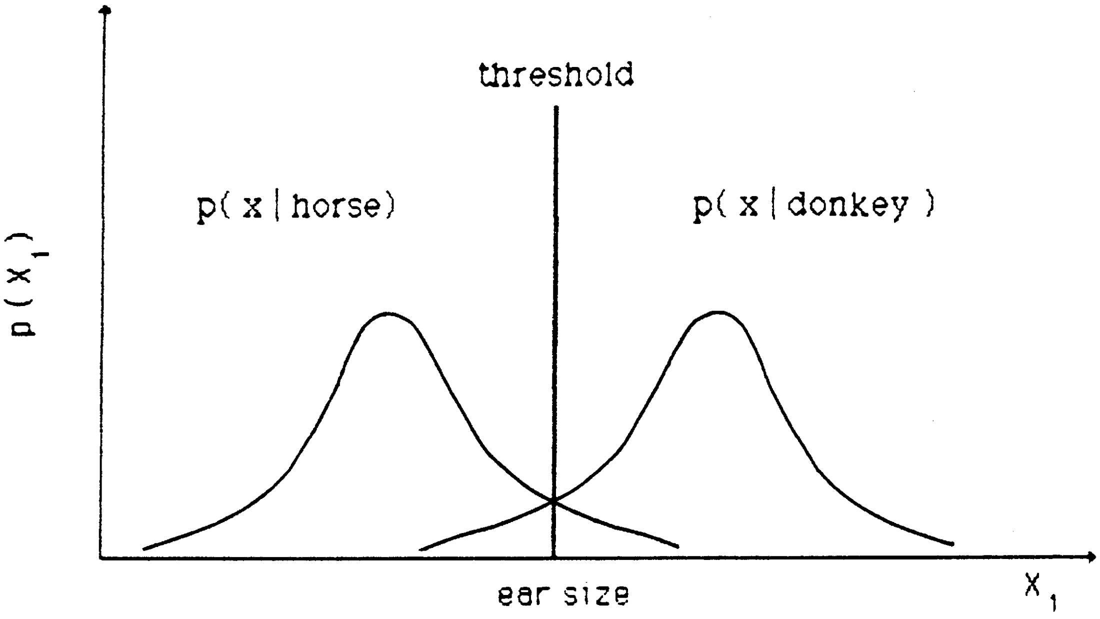


Fig 2.7 The Occurrence of a Horse or Donkey is Equally Likely

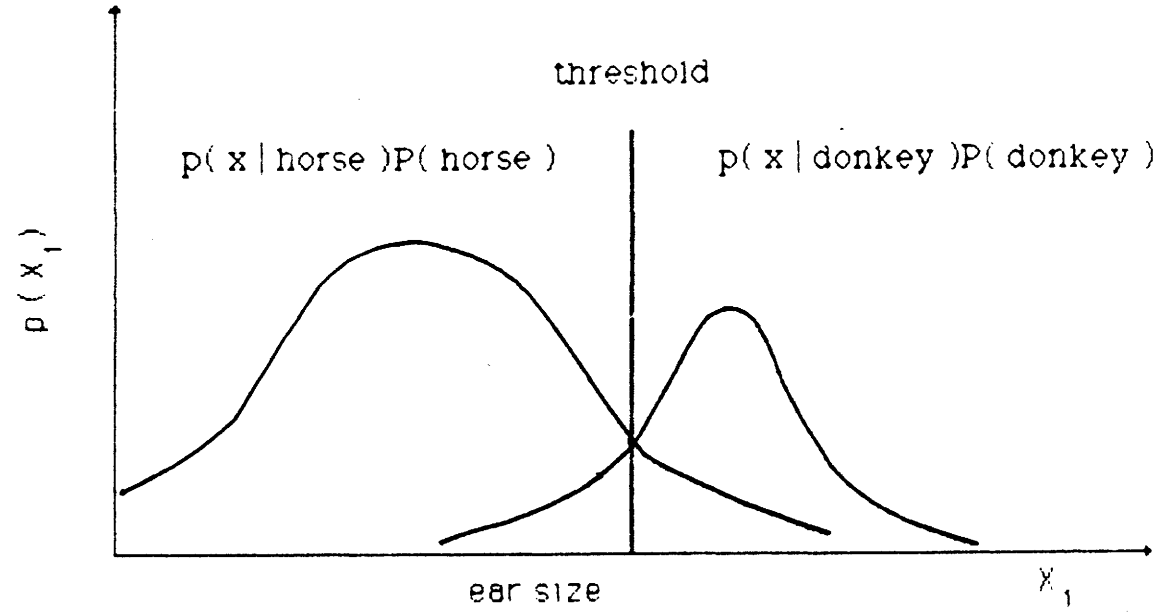


Fig 2.8 Horses are More Likely to Occur than Donkeys

Trainable Classifiers

The classification techniques discussed in the previous sections were not automatically trainable. They did not make explicit use of the learning element depicted in Figure 2.1. Coefficients for different algorithms were calculated directly from the labelled samples and substituted into the predefined formulae. "Trainable" classifiers use an iterative approach to define their decision functions. As with the other techniques, a formula exists to define the form of the decision boundary, all that is required is the coefficients. Trainable classifiers *learn* the values of the coefficients from the representative samples of each class. An assumption is made that the features selected are sufficient to define a unique boundary in *n* space that will separate all the classes.

Two approaches may be taken to define these coefficients, a *deterministic* approach or a *statistical* approach [9] [34]. The deterministic approach relies solely on the individual samples. The statistical nature of the sample classes is not used. The coefficients for the assumed function are calculated by techniques such as iteratively trying to minimize all the sample-to-boundary line distances. A common trainable deterministic classifier is the Perceptron Algorithm [18] [34].

The statistical approach uses Bayes decision functions in its classification, just as MAP did. The Bayes decision functions minimizes the probability of error. From the MAP discussion, the threshold boundary was defined as

$$p(\mathbf{x}|\text{horse})P(\text{horse}) = p(\mathbf{x}|\text{donkey})P(\text{donkey})$$

which was the point where the two curves intersected.

According to Bayes theorem

$$p(\mathbf{x}|\text{horse}) = P(\text{horse}|\mathbf{x})p(\mathbf{x})/P(\text{horse})$$

therefore the threshold may be defined as

$$P(\text{horse}|\mathbf{x}) = P(\text{donkey}|\mathbf{x})$$

The trainable statistical approach to classification differs from MAP in that MAP required estimation or knowledge of the pdfs $p(\mathbf{x}|\text{horse})$ and $p(\mathbf{x}|\text{donkey})$. In the statistical training approach, the patterns of the donkey class affect the estimation of $P(\text{horse}|\mathbf{x})$ and vice versa. This was not true in the case of the pdfs required for the MAP classifier. As a result, $P(\text{class}|\mathbf{x})$ must be learned iteratively in an interactive mode. Stochastic process are often used. The Robbins-Monro algorithm is a popular method [34].

Learning Without a Teacher

All the classifiers described thus far work for labelled data. Known sample data was given from which features could be extracted and classes formed. Unlabelled data could then be classified. What if all the data given were unlabelled? The task of classification would be much more difficult. Just as in the problem of biological taxonomy, naturally occurring classes or groups must be found without any knowledge of their number. Clustering techniques may be used to find those naturally occurring groups.

a) Clustering

A cluster can be defined as a set of samples which are similar to each other. Intuitively, a cluster should maximize between cluster distance and minimize within cluster distance when plotted in feature space. Again this yields small diameter circles/ellipses that are far apart. Reconsider the horse-donkey example, but instead of being given a herd of horses and a herd of donkeys, a corral full of mixed horses and donkeys is given. The problem is to recognize that there are two distinct classes in the given sample and to derive a method for distinguishing between the two.

Many problems are immediately apparent. How many different classes exist in this corral? What distinguishing features can be used to differentiate between the individual patterns? What method should be used to define a classifier?

There are many algorithms for clustering, hierarchical clustering, minimum variance clustering, graph theoretic clustering [18] [33] [34]. An example of a simple clustering algorithm that gives the reader an intuitive feel for the problem is

The K-Means Algorithm [18] [34]

1. Choose K means arbitrarily Z_1, $Z_2, \ldots Z_K$
2. Assign the N samples to the K clusters using a minimum euclidean distance rule from each sample to the cluster mean.
3. Compute new cluster means
4. If any of the means changed go to 2, else STOP

The efficiency of this algorithm is greatly affected by the initial selectioa of the K means.

It should be noted that clustering algorithms are notoriously slow and require a lot of storage space. Regardless, they are a workable technique for solving problems of learning without a teacher. Once classes have been isolated, any of the afore mentioned classifiers may be used.

2.3.2. Syntactic Pattern Recognition

Syntactic PR was originally developed in the 1960's, although much of the research was not done until the mid 1970's. The major difference between syntactic PR and decision-theoretic PR is that syntactic PR not only represents the pattern but also the structure within which the pattern occurs. In decision-theoretic PR each feature is used once to measure similarity within one dimension of the total pattern representation space. Features are measured quantitatively with no emphasis placed on the relationship between features. Conversely, syntactic PR explicitly uses features to build pattern primitives. The primitives combine structurally to create a pattern. This structure is explicitly defined.

Syntactic PR techniques are only applicable to a subset of the Pattern Recognition problems. In some classification problems, no relationship exists between features. An example of a domain in which syntactic PR has been used quite frequently is scene analysis, where relative positions of objects or pattern primitives within the scene are important. Another application is speech recognition.

The structural description of a pattern has been found to be analogous to the description of the syntax of a language. Consequently, many of the techniques of formal language may be applied to the pattern recognition problem. Patterns are composed of subpatterns, which in turn are composed of a set of pattern primitives. Similarly, in formal language, sentences are composed of phrases which in turn are composed of words. A *pattern description language* is a language that provides the structural description in terms of primitives and relations. The *grammar* of a pattern description specifies the rules for the composition of primitives into patterns. Patterns are classified by performing a parsing of the *sentence* describing the unknown pattern. This sentence is deemed grammatically correct or incorrect with respect to the grammar.

2.3.2.1. Knowledge Representation

As in the decision-theoretic approach, pattern primitives are building blocks in the creation of a sufficient representation for a pattern. These pattern primitives are used as terminals in a pattern grammar utilized to hierarchically represent the structure of the pattern. Depending on the type of grammar, a pattern sentence can be represented as either a string, a tree, or a graph. The primitives selected should be recognizable by decision-theoretic PR techniques. They should serve as an adequate basis, when combined with structural relations, to represent any pattern. The hierarchical approach is well suited to complex patterns. Often these patterns have many features. By dividing the pattern description into subpattern descriptions and so on down to a set of basic primitives, the description and thus classification task is greatly simplified. There are two classic examples of syntactic Pattern Recognition applications that are used in most of the literature. The following string grammar example was taken from [13], but can also be found in [14].

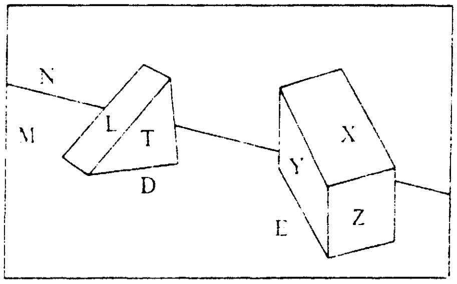


Fig 2.9 Scene A [13]

Figure 2.9 represents a picture pattern, Scene A. This pattern contains a lot of structural information and is thus a good candidate for description using syntactic PR. Figure 2.10 shows a hierarchical structural description of Scene A. The analogy between this hierarchical description and a language parse tree is quite evident. A relational graph representation of Scene A is illustrated in Figure 2.11. The reader familiar with AI knowledge representation techniques may be reminded of the semantic net. The relational graph lends itself to expression as a relational matrix and thus to techniques applicable to relational matrices. It does not allow for the use of formal language techniques.

To formalize the relationship between language theory and syntactic PR, some terminology will be introduced. A grammar G is made up of four components.

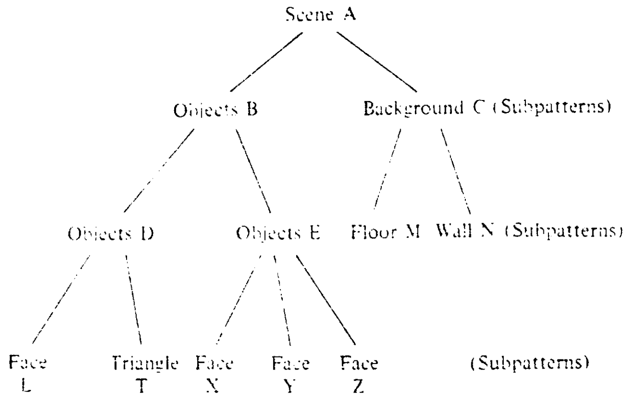


Fig 2.10 A Hierarchical Structural Representation of Scene A [13]

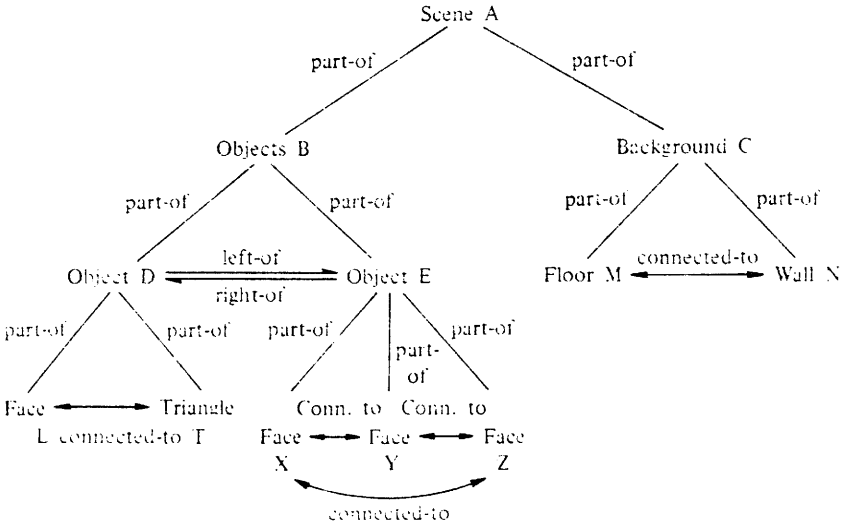


Fig 2.11 A Relational Graph Representation of Scene A [13]

1) V_N : a finite set of nonterminals

2) V_T : a finite set of terminals

3) P : a finite set of production rules

4) S : a start symbol contained in V_N

Thus the grammar G can be represented by the four-tuple

$$G = (V_N, V_T, P, S)$$

The following example illustrates the use of syntactic PR for recognizing images of specific chromosomes. The primitives selected for the description of the chromosome are portions of the chromosome, easily recognized by conventional decision-theoretic PR techniques. This example has been described in [13] [14] [34]. The task is to classify chromosomes as being either ***submedian*** or ***telocentric***.
>>>

Fig 2.12 (a) Submedian Chromosome (b) Telocentric Chromosome [13]

This classification problem has an inherent structural nature. Figure 2.13 shows how a submedian chromosome can be represented hierarchically.

Five primitive features or terminals were selected as the building blocks for the grammar. The grammar itself is depicted in Figure 2.14.

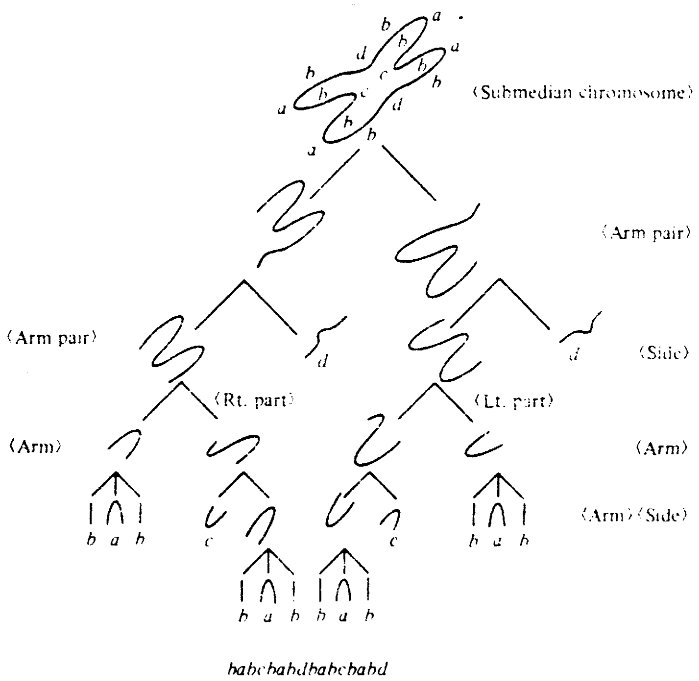
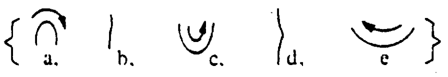


Fig 2.13 A Hierarchical Structural Representation of a Submedian Chromosome [14]

$G=(V_N, V_T, P, \{$⟨submedian chromosome⟩, ⟨telocentric chromosome⟩$\})$

where

$V_N=\{$⟨submedian chromosome⟩, ⟨telocentric chromosome⟩, ⟨arm pair⟩, ⟨left part⟩, ⟨right part⟩, ⟨arm⟩, ⟨side⟩, ⟨bottom⟩$\}$

$V_T=\{$  $\}$

and

P. ⟨submedian chromosome⟩→⟨arm pair⟩ ⟨arm pair⟩

⟨telocentric chromosome⟩→⟨bottom⟩ ⟨arm pair⟩

⟨arm pair⟩→⟨side⟩ ⟨arm pair⟩

⟨arm pair⟩→⟨arm pair⟩ ⟨side⟩

⟨arm pair⟩→⟨arm⟩ ⟨right part⟩

⟨arm pair⟩→⟨left part⟩ ⟨arm⟩

⟨left part⟩→⟨arm⟩ *c*

⟨right part⟩→*c* ⟨arm⟩

⟨bottom⟩→*b* ⟨bottom⟩

⟨bottom⟩→⟨bottom⟩ *b*

⟨bottom⟩→*e*

⟨side⟩→*b* ⟨side⟩

⟨side⟩→⟨side⟩ *b*

⟨side⟩→*b*

⟨side⟩→*d*

⟨arm⟩→*b* ⟨arm⟩

⟨arm⟩→⟨arm⟩ *b*

⟨arm⟩→*a*

Fig 2.14 A Pattern Grammar for Classifying Submedian and Telocentric Chromosomes [13]

Primitive Selection and Grammatical Inference

As with features in decision-theoretic PR, the selection of primitives is an important aspect of knowledge representation. Of additional concern in syntactic PR is the creation of the hierarchical structure and accompanying pattern grammar. The selection of primitives is application dependent. There is no generalized solution to date.

This lack of an automated, generalized solution is related to the fact that most of the applications of syntactic PR are in image processing, speech recognition or related fields. As a result, the sensing stage of the PR system is complex and can result in the transformation of a pattern into many different measurements. There are a vast number of primitives that can be selected. Conversely, in a problem like the decision-theoretic horse-donkey application, feature selection is an easier task because of the more limited, discrete measurements from which to choose.

Accompanying the selection of primitives is the creation of a structural grammar to describe the pattern class. The task of creating a grammar is very much related to the selection of primitives. The more basic the primitives, the more complex the structural grammar. Given a set of primitives, it is not a difficult task to manually design a grammar. Much work has been done in syntactic PR on what is called *grammatical inference*. This is the automatic generation of grammars from a set of labelled examples. Unfortunately, this is beyond the scope of this introduction. It is sufficient to say that no generalized techniques exist. There are numerous algorithms for inferring restricted grammars. The reader interested in machine learning is advised to see the related material in [14] [34] and the references therein.

Extensions of String Grammars

The structural representations discussed this far have involved only concatenation. High-dimensional pattern grammars [13] [14] [34] have also been created to represent two and three dimensional patterns. These result in the representation of patterns as trees or graphs. Figure 2.15 is an example of a two-dimensional pattern grammar for representing houses [12] [14]. Instead of the implied concatenation of primitives, explicit operators exist to link together primitives in various ways.

Other high-dimensional grammars include: *web grammars* [13] [14] which generate directed graphs with symbols at their nodes; *branch oriented grammars* such as Plex and PDL that generate graphs with primitives on their branches [13] [14];

$G = (V_N, V_T, P, S)$

where

V_N = {⟨house⟩, ⟨side view⟩, ⟨front view⟩, ⟨roof⟩, ⟨gable⟩, ⟨wall⟩, ⟨chimney⟩, ⟨windows⟩, ⟨door⟩}

V_T = {▯, ▯, ▯, ⊞, △, □, ⏢, →, (•), ⊙, ○, ↑, ↦}

S = ⟨house⟩

P: ⟨door⟩ → ▯

⟨windows⟩ → ⊞, ⟨windows⟩ → → (⟨windows⟩, ⊞).

⟨chimney⟩ → ▯, ⟨chimney⟩ → ▯

⟨wall⟩ → □, ⟨wall⟩ → ○ (⟨door⟩, □)

⟨wall⟩ → ⊙ (⟨windows⟩, □)

⟨gable⟩ → △, ⟨gable⟩ → ↑ (⟨chimney⟩, △)

⟨roof⟩ → ⏢, ⟨roof⟩ → ↑ (⟨chimney⟩, ⏢)

⟨front view⟩ → ↑ (⟨gable⟩, ⟨wall⟩)

⟨side view⟩ → ↑ (⟨roof⟩, ⟨wall⟩)

⟨house⟩ → ⟨front view⟩

⟨house⟩ → ↦ (⟨house⟩, ⟨side view⟩)

The notation

→ (X, Y) means that X is to the right of Y

⊙ (X, Y) means that X is inside of Y

○ (X, Y) means that X is inside on the bc:tom of Y

↑ (X, Y) means that X rests on tops of Y

↦ (X, Y) means that X rests to the ıight of Y

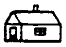
Housc	Description
	↑(△,□)
	↑(↑(▯,△),○(▯,□))
	↦(↑(↑(▯,⏢),⊙(⊞,□)),↑(△,○(▯,□)))

Fig 2.15 A Two Dimensional Pattern Grammar [14]

attributed graphs [35] [36] which generate graphs with attributes on the nodes and branches; and *tree grammars* that extend 1-dimensional concatenation to multidimensional concatenation, resulting in the production of trees instead of strings or graphs.

A common problem in both decision-theoretic and syntactic PR is overlap in class representation. This can be visualized as intersecting clusters in the decision-theoretic approach. The overlap is often caused by either noise in measurement data, or by representing classes with features that inadequately capture the information required for full differentiation. This can usually be overcome by the use of probability and statistics. In syntactic PR, a pattern grammar,

$G = (V_N, V_T, P, S)$

can be transformed into a *stochastic pattern grammar*

$G = (V_N, V_T, P, Q, S)$.

This is done by assigning a probability measure to each of the productions of P. The probability measures are represented by the set Q. Techniques exist for learning the production probabilities from examples. These techniques are not fully generalized for all types of grammars. A brief explanation is found in [33].

2.3.2.2. Classification

A set of *M* pattern classes may be represented by *M* or less distinct grammars. If classes are similar, it is often more efficient, as in the chromosome example, to combine grammars and to have two separate start symbols, depending upon the class. Most of the classification techniques in syntactic PR are based on the formal language concept of parsing. Both top-down and bottom-up parsing may be used. Top-down parsing involves taking the start symbol and from left-to-right, trying to break it down into a sentence of the grammar by successively applying production rules to produce terminals. Conversely, bottom-up parsing starts with the sentence of terminals and successively applies the production rules backwards until the start symbol is reached or it is decided that the task is impossible.

By looking at the chromosome grammar, it is evident that there are decisions to be made as to which production rule to apply at a particular point in the parsing procedure. To avoid an exhaustive search, and to minimize backtracking, heuristic rules are often added to the search process to assist the decision-making process. When dealing with stochastic languages, the production probabilities can be used as heuristics in the search strategy. The production offering the highest probability of success should be tried first.

An advantage of structural descriptions and parsing for classification problems is that it is easy to detect specifically where a pattern differs from a class grammar. The structural description can provide a limited explanation of why a pattern does not belong to a specific class.

Error-Correcting Parsing

Stochastic languages attempt to minimize misclassification of noisy or distorted patterns by enhancing grammars with probabilistic information. In decision-theoretic PR, the concept of *similarity*, as opposed to perfect equality was used extensively to combat this problem. Similarity measures have been extended to syntactic PR via *error-correcting parsers*. The criterion for similarity is often minimum distance. The distance between two strings, or between a string and a language can be described in terms of error transformations. There are three transformations that can be performed on a string to cause it to deviate from a sentence of a language. The three transformations are: *substitution*, where one primitive of the class is substituted for another primitive also belonging to the class; *deletion*, where a primitive occurrence is deleted from the string; and *insertion*, where a primitive of the class is inserted into the string. The number of transformations required to convert a deviant string into a string of a language constitutes that pattern's distance from that class. Transformations can be weighted if some types of errors are less likely than others.

Some error-correcting parsers add the three types of error transformations to their grammar. Production rules are created to represent all possible strings generated by these errors. A string is then classified as belonging to the class which is the a minimum distance from that string. Just as there are several different measures of similarity in decision-theoretic PR, there are several types of error-correcting parsers.

Clustering

The notion of cluster analysis, used extensively in decision-theoretic PR, has been extended to deal with problems in syntactic PR. Not only must the similarity of the primitives be taken into account, but also the similarity of the resultant pattern structures. Classical clustering algorithms such as minimum spanning tree and K-means have been enhanced for syntactic PR problems. These algorithms have been combined with error-correcting parsing and grammatical inference techniques to generate a grammar for a set of patterns [14]. Unfortunately, a detailed description of syntactic clustering would require a more in depth background than provided herein. Further study in this area is recommended.

3. A Comparison of PR and AI

3.1. Interaction Between PR and AI

As stated previously, there has been little interaction between the fields of PR and AI. Most of the literature combining facets of the two fields, deals with the application of AI techniques to PR systems. These systems almost exclusively perform the task of scene analysis or speech recognition. AI techniques are used for high level semantic processing [25], whereas PR is suited to manipulating low level primitives, which These do not necessarily map to real world knowledge. AI can represent knowledge at any level. Its application to represent knowledge about knowledge, *metaknowledge*, assists in the reasoning strategy and makes use of human expertise at performing a task. PR has no efficient mechanism for representing and utilizing such knowledge.

The AI methodology can be applied to a PR system and remain intact. The converse is not true. There is little mentioned in AI/PR literature of the application of PR techniques to AI systems. PR is a toolbox of techniques. Tools exist to assist AI systems in the selection of knowledge, decision-making, and learning. There have been some AI researchers in the past who have realized the potential of Pattern Recognition. Banerji, in **Theory of Problem Solving** (1969) [1] tried to convince his readers of the value of PR techniques. Other work in AI appears to have reinvented some of the methods of PR. This could have been prevented by better communication between the two fields. Early AI work in learning by Samuel [6] makes use of the concept of features and adaptive learning. More recent work by Buchanan, Mitchell, Michalski [21] [22] [23] and Rendell [29] [30] indicates the use of PR as a tool in AI machine learning. Much more work is needed in the application of Pattern Recognition principles to Artificial Intelligence.

3.2. Metrics for Comparison

3.2.1. Methodology

There is a basic difference in methodology and approach taken by AI researchers as compared to PR researchers. Central to Artificial Intelligence is the ***physical symbol system hypothesis*** proposed by Newell and Simon in 1976 [31].

> *"A physical symbol system consists of a set of entities, called symbols, which are physical patterns that can occur as components of another type of entity called an expression (or symbol structure). Thus, a symbol structure is composed of a number of instances (or tokens) of symbols related in some physical way (such as one token being next to another). At any instant of time the system will contain a collection of these symbol structures. Besides these symbol structures, the system also contains a collection of processes that operate on expressions to produce other expressions: processes of creation, modification, reproduction and destruction. A physical symbol system is a machine that produces through time an evolving collection of symbol structures. Such a system exists in a world of objects wider than just these symbolic expressions themselves."* [31]

The *physical symbol system hypothesis* is:

> *"A physical symbol system has the necessary and sufficient means for general intelligent action."* [31]

This hypothesis has not yet been proven or disproven. Artificial Intelligence attempts to support its validity by the application of symbolic computing techniques to the performance of intelligent human tasks by a computer. This involves representing knowledge as symbol structures and manipulating this knowledge using search strategy. Knowledge, its acquisition, representation and manipulation is at the heart of all intelligent tasks. These three areas have been isolated for more in depth discussion in subsequent chapters.

As stated in Chapter 1, one of the original motivations for the AI methodology was to model human thinking on a computer, to create computer programs that perform human tasks the way humans do them. This has been part of the motivation for the physical symbol system approach to computing. By using symbol structures to represent knowledge, a direct mapping can be established between the real world model and the computing model. While symbolic computing is the major implementation mode for most AI applications, not all of today's AI applications model human thought. In recent years, the AI goal has been to perform tasks that humans perform, in the easiest way possible.

Decision-theoretic PR and syntactic PR may be compared; they are sufficiently different in methodology. Syntactic PR is built on top of decision-theoretic PR primitives, so the comparison of PR to AI may begin with decision-theoretic PR. Decision-theoretic PR does not represent its problems as physical symbol systems.

This is a fundamental difference in methodology. To reiterate comments made in Chapter 1, the main goal of Pattern Recognition researchers in the 60's, as today, was to make economical, efficient, workable systems. To do so, they based their techniques on Von Neumann programming principles, using what then was present day technology. The result today is a well established discipline, able to solve many machine intelligence problems within specific domains. This methodology has its limitations though. There is little relation between the Pattern Recognition representation of a problem, and the real world model. As such many PR systems cannot benefit from expert human knowledge.

Syntactic PR is a step towards the physical symbol system approach. Subpatterns in a pattern description are represented as symbols, physically related in a symbol system structure or hierarchy. However, the basic primitives, and even the subpatterns are selected on the basis of their suitability for nonsymbolic processing. They do not necessarily map to the real world model. This contradicts AI methodology. The inability of Pattern Recognition systems to represent and manipulate higher level world knowledge, has prompted the application of Artificial Intelligence techniques for high level semantic processing. It has enhanced several PR systems [25].

3.2.2. Formalization

Although PR techniques are applicable to many different domains, the field has focussed on a subset of these application areas, the use of computers to give machines sensory capabilities. Consequently, there are specific techniques for speech recognition problems, computer vision problems etc. These are established methods for application to specific domains. The methods are affected by the characteristics of the data. As a result human expertise is sometimes required for fine tuning of PR systems.

Conversely, AI has tried to establish generalized techniques for creating computer programs that perform human tasks. There are basic knowledge representation and problem solving techniques. However, for many of the large AI applications such as expert systems, domain specific fine-tuning prevails. AI programs are heavily customized for the specific application. Only now are more established methods for building AI programs being created. This is a reflection of the size and diversity of the problem of generalized machine intelligence. Unfortunately, it makes comparison of specific AI and PR principles, such as complexity, difficult.

3.2.3. Implementation

As a metric for comparison, *ease of implementation* may be divided into two subcategories, *manpower requirements* and *software, hardware and equipment requirements*.

Large AI applications, such as expert systems, have far greater manpower requirements than PR systems. Many of the expert systems to date have required man-decades to implement. Again this is a reflection of the lack of formalism in techniques and the nature of the specific applications to which AI has been applied. The major drain on manpower is in the acquisition and sufficient representation of knowledge. Computer professionals and domain specific experts must combine forces to select required knowledge and a means by which to represent it. Extensive work must then be done to ensure a satisfactory level of performance. The task would be reduced immensely if there were one professional, trained in both areas or if automatic knowledge acquisition could be performed by the computer. AI programs are often more sophisticated and user-friendly than PR systems. Expert systems have user interfaces and facilities for providing explanations. More development time is required to add all these tools.

In PR, especially decision-theoretic PR, the task of knowledge acquisition and representation is more automated. Knowledge required for the decision-making process is *learned* by the computer on the basis of labelled training samples. A human is required to collect the training samples and to assist in the sensing stage (see Fig 2.1). He/she may also be needed for assistance in feature selection, depending upon the problem domain. The role of the human is far less significant than in an AI application. In decision-theoretic PR there is no explicit mapping with the real world model; knowledge is represented in a mathematical form that is easily manipulated by Von Neumann programming techniques. Therefore a human expert is not needed to map real world knowledge into a computer framework. Furthermore, there is no interaction between the user and the computer; hence, there is no need to design a user interface.

In syntactic PR the knowledge acquisition and representation process is less automated. Although primitive selection is not fully generalized, because of the limited domain of applicability of syntactic PR, primitives have been substantially preselected for specific domains. Hence, little human interaction is needed. Grammar generation from labelled samples is not fully automated. The structural representation used by the PR system does not have to map to one used by an expert. Therefore, generation by a human is a minor task. The manpower requirements for a syntactic PR system are still significantly less than for an AI system.

Software, hardware and equipment requirements are very different for PR and AI systems. AI operates in a symbolic programming environment. Specialized AI languages such as LISP and PROLOG are needed. There are also many software tools available for specific AI tasks. For example, planning programs such as STRIPS and expert system shells such as EMYCIN and OPS5 are frequently used. The software environment is generally high level and, in agreement with AI methodology, user-friendly. AI tasks have been coded in conventional Von Neumann programming languages such as C and Pascal. AI purists might argue however that these are not AI applications because they lack the segregation of knowledge base and inference engine characterizing an AI program.

At present, AI software runs on Von Neumann machines. Processing is sometimes slow. Recently, a machine was designed and marketed specifically for processing LISP code. Similar machines are being designed for PROLOG. In the future, AI may have specialized hardware, unless the AI programming methodology becomes the norm.

PR has fewer specialized software requirements. It utilizes Von Neumann programming techniques and thus can use conventional programming languages. Because of the mathematical nature of decision-theoretic PR techniques, languages that are capable of rapid mathematical calculations, high precision, and calculations on *n* dimensional matrices are desirable. Von Neumann machines are capable of running decision-theoretic PR programs, albeit slowly. Array processors or machines with distributed storage and processing are faster and more efficient because of their specialized architecture. When dealing with syntactic PR systems, a compromise must be made between facilities for numeric calculations and facilities for parsing of pattern grammars.

Pattern Recognition systems have additional equipment requirements because of the problem areas to which they have been applied. To give a machine sensory capabilities, equipment is needed to accept the original sensory stimulus and to transform it into a machine useable form. For example, computer vision systems require photographic equipment and digitizers. Speech recognition systems need acoustic equipment to generate sound spectrograms.

3.2.4. Understanding and Modification

An important component of any piece of software is the ability to understand how it works and to be able to modify it if necessary. AI programs are much easier to understand and modify than PR programs. The investment of time in the development stage to create a mapping between world knowledge and the internal knowledge of the program pays off in this regard. Also the fact that the knowledge base and control structure are separate, allows for an updating of knowledge without affecting the search strategy. If knowledge is organized hierarchically, metaknowledge r heuristics can be added to enhance the system. Similarly, different search strategies may be tested to try and optimize the system without affecting the knowledge store.

PR systems are much more difficult to understand and to modify. As knowledge and inferencing are interwoven. A programmer can comprehend the mathematical algorithms being performed. However, it is much more difficult to see the correlation between the *n* dimensional vectors and matrices, the mathematical formulae and the human task that is being performed. As such, it is difficult for a human to modify the program unless it is simply to add another feature at the feature selection stage, or to select another classification algorithm. There is no obvious cause-and-effect relationship between what goes into the PR system, and what come out. Therefore it is difficult to modify a PR program on the basis of a specific performance inadequacy.

3.2.5. Application Domains

It may be argued that PR and AI need not be compared because they are applicable to different machine intelligence problems. AI does not have the mathematical ability to analyze digitized scenes or sound waves. Similarly, decision-theoretic PR techniques are not well suited to storing knowledge such as *red* or concepts such as *love*. They cannot solve problems such as those encountered in Natural Language Understanding. Syntactic PR might be altered to address some of these symbolic problems. However, it is incapable of dealing with them as it exists.

PR is a generalized classification/decision-making technique, dependent on the character of the data to be analyzed. A PR system selects criteria for the decision-making process on the basis of training samples. It is thus well suited to *any* problem domain that generates a lot of data and requires decision-making or classification. PR systems have been designed in such diversified problem areas as medical diagnosis [3] [4] and the granting of credit [5].

Unlike AI, PR can perform human tasks whose functioning is not fully understood. This is a very powerful quality. It is one reason why PR is so well suited to problems of sensory perception. In such problem domains, performance algorithms and heuristics cannot be defined by an expert. Few AI expert systems have powerful learning elements. Consequently all knowledge must be supplied by a human expert. The resulting system is only as good as the expert providing the original knowledge.

PR systems are capable of dealing with inexact data. Decisions made using measures of similarity as opposed to equality, allow for both noise and outliers. This facility has been sorely lacking in Artificial Intelligence systems.

The PR approach is applicable to any domain where classsification or decision-making is the main task. The domain must generate sufficient data for mathematical analysis, and must be representable using the limited decision-theoretic KR techniques.

Syntactic PR, without modification, is restricted to problems that have physical structural characteristics and that are built of decision-theoretic primitives. Scene analysis and speech recognition are two such problem domains. Beyond this, there are few applications for this specialized technique.

The greatest asset of the Artificial Intelligence approach is its ability to represent knowledge and heuristics. Human tasks that require conceptual knowledge and rules of thumb are better suited to AI methodology. Artificial Intelligence systems provide a user-friendly, interactive environment that is suitable for many human problems where the required input is not known in advance. Hence, AI is applicable to a much broader range of problem domains.

3.2.6. Future Research Potential

Both Pattern Recognition and Artificial Intelligence have many unresolved problems. There is room for work in both areas. PR is a well established discipline. None of the questions are new. It is possible that PR techniques alone are not capable of solving its problems.

Each stage in the PR process has unique issues. The sensing stage could be improved by the introduction of more sophisticated sensing equipment. For example, low level lasers are being used to extract measurements in some experimentary computer vision systems. This can yield vital depth information without extensive calculations.

Feature selection/extraction is a fascinating problem and would be an extremely powerful tool if it were it fully automated. Decision-theoretic PR methods have provided a limited solution. Primitive selection in syntactic PR is not automated either. Currently, syntactic PR is applied to such a narrow range of problem domains that primitives can be very easily preselected manually. This is not a final solution. Grammatical Inference is directly related to learning work done by AI researchers such as Winston [6]. There are many possibilities for expansion of present work in this area.

In both syntactic and decision-theoretic PR, an ideal classifier would characterize the data fully. Pdf based, decision-theoretic classifiers have minimized the probability of misclassification of a pattern. More complex alterations on the decision-theoretic techniques are proposed yearly. Work in stochastic grammars and error-correcting parsers in syntactic PR has great potential. Clustering techniques are already well developed for decision-theoretic PR. More research should be performed in syntactic clustering. To summarize, decision-theoretic PR is reaching a plateau adhering to a strict PR framework. Some syntactic PR problems are related to AI problems. Hence, research in this area may benefit the field of AI.

AI has much room for expansion. Even adhering strictly to current AI methodology, there are many areas where research has just begun. One major task is to design a knowledge representation scheme that is sufficiently general for most domains. This knowledge representation scheme would need to be combined with a powerful inferencing mechanism. Machine learning is an important area of research in AI. To date it has had only limited and very specialized success. The work on symbolic processing machines will stimulate work on new symbolic programming languages and accompanying software tools. Both in Pattern Recognition and in Artificial Intelligence, it would be a great asset to have more generalized techniques. This would make individual machine intelligence applications much easier to perform.

Moreover, the develpment of composite AI/PR systems or tools warrants further attention. This issue is addressed in the following three chapters. Techniques in PR are compared to current techniques in AI. Suggestions are made as to where PR tools might fit into the AI framework.

4. Knowledge Representation

For a computer to display intelligent behaviour, it first needs a method for acquiring and storing knowledge. The representation of this knowledge must be compatible with a method for its manipulation. The following discussion of PR and AI techniques for Knowledge Representation has been divided into two subcategories: *Computer Representation* and *Knowledge Selection*. Computer Representation, more conventionally known as Knowledge Representation in most AI literature, is the task of physically representing knowledge in a computer. Knowledge Selection is the initial job of taking a great deal of information on a problem domain and selecting a concise, sufficient subset of that information to represent the domain. Knowledge selection is a subset of knowledge acquisition discussed in Chapter 6. Representation not only implies the physical method by which something is described, but also the conceptual portrayal of that notion. The selection of content or information used to characterize a concept is crucial to the performance of any machine intelligence program.

4.1. Computer Representation

The AI term *Knowledge-based system*, puts proper emphasis on the importance of knowledge in the implementation of a machine intelligence program. There are several characteristics of knowledge that restrict the choice of a computer representation scheme. A computer can only handle a limited store of information; hence, knowledge must be represented concisely, and restricted to a specific domain. A computer representation scheme should not be rigid; knowledge often changes with time. Knowledge can be expressed using generalizations and analogies. Furthermore, knowledge about knowledge exists and can be used to alleviate the burden of managing knowledge. Ideally, a computer representation scheme would include facilities for handling all of these features.

4.1.1. A Comparison of PR and AI

The computer representations used by decision-theoretic PR and AI epitomize the polarity in these methodologies. The fundamental difference is that Artificial Intelligence uses the physical symbol system approach. Knowledge is represented in terms of symbols and symbol structures. These symbols can map directly to real world knowledge. Pattern Recognition represents knowledge in terms of features with pattern vectors, covariance matrices and probability density functions. The features are a mapping of world knowledge into a minimum representation. The features do not necessarily have an intuitive meaning. The pattern vectors, matrices and pdfs are used to create a classifier that performs the decision-making. Knowledge and the inference mechanism are intertwined. Conversely, the AI approach is characterized by a separate knowledge base and inference mechanism.

Syntactic PR is built on the same nonsymbolic foundation as decision-theoretic PR. Knowledge is represented at an elemental level by a set of primitives that are identified using decision-theoretic techniques. These primitives are selected on the basis of ease of recognition and do not correspond to an expert's set of primitives. Primitives are combined to form subpatterns, and so on, up to a pattern. The subpatterns do not necessarily correspond to logical divisions in a pattern. The hierarchical description of a pattern class as a combination of subclasses and primitives is represented using a language grammar. The language grammar is symbolic in nature but lacks the conceptual mapping of an AI scheme. It is reminiscent of a semantic net in appearance.

Pattern Recognition does not have a very powerful computer representation scheme by AI standards. The concept of knowledge is more fundamental to the AI approach. PR is better judged as a whole system than as the sum of its parts. In support of the PR approach, it could be argued that knowledge symbols such as *blue* or *small* mean nothing to the computer. They also yield no common ground for comparison. Conversely, PR uses low level primitives or features. All features are comparable and measurable within the common medium of feature space. Low level primitives are building blocks, especially in the area of machine learning. It is often from low level primitives that new concepts can be extrapolated. The fact that the primitives do not always have intuitive meaning should be secondary to the fact that they generate a successful method for classification. The human thinking model is only one way of performing a task, and not guaranteed to be the most efficient.

All this is not an argument for abandoning the AI approach. It is much more flexible than the PR approach. The AI computer representation maps to real world knowledge. This makes it understandable, easily modified and capable of generating explanations of its reasoning. PR knowledge is translated, condensed and intertwined with its inferencing mechanism. Therefore, it is difficult to understand and not easily modified. Decision-theoretic PR cannot yield an explanation of how or why it made a decision. The hierarchical structural description in syntactic PR gives it more power. It can at least tell where in the hierarchical structure of a class, an unknown pattern deviated. The AI approach can represent many different types of knowledge. Concepts such as "likes", "hates", descriptors such as "red", "small", "very", and numeric data can all be represented. PR techniques can only represent numeric measurements or concepts that can be mapped to a numeric scale as primitives. A very important attribute of the AI scheme is its ability to represent metaknowledge and heuristics. These two types of knowledge greatly assist in the decision-making process. PR has neither the capability to represent metaknowledge nor the capability to represent heuristics, yet these are needed in many PR applications.

The PR approach does have some advantages. The most important of which is its ability to represent inexact knowledge including noisy data, incomplete data and outliers. PR uses a representative sample of a class to formulate a description of that class. When possible, it uses a pdf to capture allowable variability in patterns. Conversely, AI represents the general case description of a class. Heuristics are used to help with common deviations. Thus, the PR approach does not require heuristics in many cases. PR condenses its knowledge thus reducing the dimensionality of the decision-making problem and highlighting important characteristics of the problem domain. Lastly, the AI approach has no generalized computer representation that works in all instances. However, neither does PR, as the selection of a classifier is data dependent. PR does have a finite number of workable schemes that can be implemented.

4.1.2. Application of PR Techniques to AI

Computer representation is not an area where PR can offer much assistance to AI. Instead, the converse is the case. Semantics, represented in AI systems, are sorely lacking in PR systems. PR computer representation does have a few attributes which would be helpful to an AI computer representation scheme. Because of the interwoven nature of knowledge and inference in PR systems, some tools will be discussed in the chapter on problem-solving. The ability to characterize variability in knowledge is a strength of PR systems. It allows for the measure of similarity as opposed to equality between input and a knowledge base. To see how PR can be applied to AI computer representation schemes, several specific schemes will be analyzed.

Logic is the classical computer representation scheme. It has been a popular representation for problems in theorem proving. Logic is one area where researchers have already undertaken to incorporate PR techniques. Banerji [1] saw a direct relationship between solutions to problems and games, and Pattern Recognition. Using Banerji's terminology, patterns are sets of objects, and objects belong to the "universe". Given an element of the universe, it may be classified as belonging to a pattern by the application of a procedure. These procedures are a set of statements, called predicates. The predicates are thus descriptions of the pattern. Banerji created a model for a description language of patterns using logic. The model was generalized and implied the embedding of specific description languages for specific types of patterns. Each description language consisted of initial predicates and means by which to connect them. Connection of the initial predicates yielded descriptions of patterns. Banerji's work was extended by Cohen who designed CODE, a logic-based description language [7].

Another researcher who applied PR concepts to a logic representation was Michalski [21] [22]. Michalski created a variable-valued logic calculus called VL_{21}. VL_{21} is an extension of predicate calculus which has been used for inferring descriptions of classes from examples or from partial class description. VL_{21} is capable of representing pattern descriptors as variables, functions or predicates. These descriptors can have several different types which Michalski described as nominal, linear and structured. Hypotheses, data rules, problem knowledge rules and generalization rules can all be described in VL_{21}. It uses selectors instead of predicates. The selectors specify relations between atomic functions. VL_{21} expressions can have a value of true, false or unknown. VL_{21} and its other version VL_1 allow the basic PR methodology of inductive inference, classification and clustering to be applied to a broader range of machine intelligence problems.

Finally, fuzzy logic [17] should be mentioned for completeness. Fuzzy logic represents an imprecise description of a piece of knowledge as a fuzzy set. Each value in the fuzzy set has a corresponding probability. For example, animal height is represented by the variable x_2 in the horse-donkey problem. The knowledge that ***horses are tall*** would be represented by a fuzzy set similar to the following:

$0 < x_2 < 4$ (feet)	probability = 0
$4 < x_2 < 5$	probability = 0.3
$5 < x_2$	probability = 0.7

Although it is not directly related to Pattern Recognition, it applies probability and variability to knowledge just as PR has.

Procedural Representation is a computer representation scheme where knowledge is contained in procedures. These procedures perform calculations or inferencing to yield knowledge. A procedural representation would be an ideal way of inserting PR-type knowledge into an AI system. The procedure name would describe what the procedural representation did but not how it did it. The low level mathematical calculations would be hidden. The reasoning of the system as a whole would still be understood. This idea could also be extended to *Production Systems*, where knowledge is represented in terms of productions rules. A production rule is an IF *condition* THEN *action* statement. Some production rules could be translated into PR procedures. For example, IF *big ears* AND *short* THEN *donkey* could be replaced by something similar to the classifiers discussed earlier. PR techniques would be helpful in discerning concepts like *big*, *tall* or *small* within the range of the domain because they are relative and variable. For both production systems and procedural representation, condensed low level features could be used to represent some knowledge. Explicit rules could be represented at a higher level. The PR classifiers would be invisible to the normal inferencing mechanism. The insertion of PR classifiers would be most effective for procedures that are static and need not be changed. Its implementation would require a programming language with powerful mathematical capabilities, as well as symbolic processing.

Semantic nets are the final computer representation to be discussed. Their structure is analogous to the structural representation used in syntactic PR. As such, some of the techniques used in syntactic PR may be applicable. Semantic nets are represented by nodes and arcs. The nodes contain entities and the arcs represent relationships between the entities represented at the nodes. Conceptually, a semantic network may be thought of as a graph or network. However, when the semantic net is actually represented in an AI program it is often represented as an attribute-value memory structure. Syntactic PR grammars have the capability of representing patterns as strings, trees, or graphs. High-dimensional grammars such as web grammars or graph grammars [14] could easily be used to represent semantic nets. This would characterize the graph-like structure explicitly in a set of productions, as opposed to implicitly. The blocks world is a domain commonly used for testing AI machine learning techniques. It is well suited to representation by syntactic PR, as illustrated in Chapter 2. This would allow for the use of syntactic PR machine learning techniques to be easily integrated into an AI framework.

The actual grammar representation is secondary to the tools that can be applied to the grammar. Techniques such as error-correcting parsing, measures of similarity as opposed to equality, graph isomorphisms and grammatical inferencing could all be utilized. Semantic nets might also be extended by the use of attributed graphs [36] [37] which allow for bidirectional branches.

Another syntactic PR technique applicable to semantic nets is stochastic languages. Stochastic languages assign probabilities of success to different production rules. This allows for the representation of uncertainty in the problem solution. It also allows for the exclusion of erroneous knowledge that may be inferred by the computer representation scheme. There are PR techniques for inferring stochastic languages and their probabilities [14].

Finally, it should be noted that the above discussion of the application of syntactic PR techniques to semantic nets could be extended to any AI representation scheme utilizing graphs or trees.

4.2. Knowledge Selection

The initial acquisition of knowledge is a vital task in the creation of any machine intelligence program. Knowledge selection is one portion of knowledge acquisition. It involves taking a large source of information on a problem domain and selecting a representative subset for the knowledge-base. The original domain information usually comes from a human expert, the designer's personal knowledge, textbooks, or representative test cases of the task to be performed. Ideally, it would be nice to store all available information, and have the computer disregard anything redundant or irrelevant. Unfortunately, computer representation schemes already require time consuming manipulation techniques. It is important to find a minimum representation for the domain knowledge. Also, without careful preselection of knowledge, there is more chance of error and side effects in the knowledge-base. The following discussion deals specifically with the selection of knowledge. It is partially motivated by the existence of PR techniques to perform this task. A broader range of knowledge acquisition facilities will be discussed in Chapter 6.

4.2.1. A Comparison of PR and AI

Pattern Recognition has many automated techniques for assistance in the selection of knowledge. AI researchers predominantly rely on human expertise for knowledge selection. This is partially a reflection of the domains to which PR and AI are commonly applied. PR often attempts to perform sensory perception. There are no experts available that can tell a system designer how to perform this task. Therefore, the system must figure out an alternative method based on the available data. Conversely, most AI applications deal with tasks that humans know how to perform. Human experts exist to assist in the selection of pertinent knowledge.

AI uses the combination of domain experts and knowledge engineers almost exclusively in the selection of knowledge. Although there are AI tools to assist in the organization and thus selection process, the task is time-consuming. There are editors and interfaces such as EMYCIN, KAS and RLL [17], explanation facilities such as EXPERT [17], and knowledge-base revision systems such as TEIRESIAS [17], which check for inconsistencies in the knowledge-base and can propose rules. None of these tools are capable of actually selecting knowledge on their own.

In recent years, there has been some work in automatic knowledge acquisition by AI researchers. Two such systems are META-DENDRAL by Buchanan [6] [8] and AQ11 by Michalski [8] [22] [23]. META-DENDRAL operates in the domain of mass spectrometry. It proposes and selects fragmentation rules for organic structures. Experimental data is analyzed to generate and test these rules. AQ11 has formulated rules for the diagnosis of soybean plant diseases as well as for the classification of microcomputers. Both systems infer general class rules from training samples using inductive inferencing techniques. This is conceptually similar to the PR approach of training from labelled samples. However, the technique is different. Both of the above programs use positive and negative training samples. Common features are searched for in the positive training samples; these features must be capable of distinguishing the positive examples from the negative examples. Both systems are, in effect, generating generalizations. In contrast, PR systems often select features on the basis of maximum difference in features.

Most of the AI knowledge acquisition programs require domain specific knowledge. Often it takes just as much time and just as many human resources to add the domain specific knowledge and fine tune the system as it would for an expert to sit down and manually select the knowledge. Another problem with these automatic knowledge acquisition systems is that they make many generalizations from the given examples, but they still have weak knowledge selection skills. They are not capable of chosing the most appropriate generalization rules and thus features. These programs are beneficial if no expert is available. However, at this point in their development, PR techniques are in general more powerful.

PR systems often have no human expert from whom to obtain domain specific knowledge. Their knowledge acquisition and knowledge selection relies on learning from training samples. The knowledge selection task in a decision-theoretic PR system is referred to as feature selection. Feature selection involves taking a finite number of measurements from training samples and translating them into *n* features. The purpose of feature selection is to reduce the dimensionality of the knowledge representation and thus to reduce the complexity of the classifier. Feature selection is not totally automated. However, a series of tools exist which may be applied to PR systems to assist in selecting knowledge. Syntactic PR requires the selection of primitives and the inferencing of a grammar. Primitive selection is a function of feature selection at a higher level. No automated techniques exist. Grammatical inferencing will be discussed in Chapter 6.

Unlike the AI approach, PR knowledge selection involves the use of mathematical measures of independence, correlation and probability. It does not perform symbolic generalizations as most AI approaches do. PR may be assisted by a human, familiar with the problem domain, who may preselect certain features before the feature selection stage.

One criterion for feature selection is ***minimum intra-class distance***. This is analogous to minimum variability in a feature. This characteristic is beneficial for knowledge selection where one class exists. It does not reflect any comparative information between classes. Another feature selection criterion is ***maximum inter-class distance***. This ensures that a feature has maximally different values in *K* different classes. A third algebraic and statistical approach to feature selection is to find an ***orthogonal set of basis vectors*** that represent as much of the variability in the knowledge as possible. Probability techniques may also be used. If a feature has a probability of occurring equal to 1, it will always occur and is not a distinguishing feature. Similarly, if two features are independent, then their individual probabilities equal the sum of the probability that they both occur.

It is obvious from this discussion that automatic knowledge selection techniques are very different in PR as compared to AI. Both use training samples as a means by which to extract knowledge. PR is more formalized and powerful as an independent technique. Human experts use some of these criteria implicitly in their knowledge selection. Unfortunately, they cannot always see the subtle correlation and independence characteristics between features, evident when sophisticated probabilistic and statistical techniques are applied. Some of the concepts could be applied to Artificial Intelligence.

4.2.2. Application of PR Techniques to AI

Feature selection, factor analysis and other related areas would have great implications were they fully automated. Many of the concepts developed thus far could be integrated into an AI knowledge selection or knowledge acquisition system. AI programs are often slow. This is a result of the huge knowledge base that must be searched. Formalized knowledge selection would be of great assistance in removing redundant information not always obvious to the expert. AI uses different computer representation schemes than PR. Some of the feature selection techniques require the mathematical nature of feature space. Many symbolic features can be translate to a mathematical scale. Concepts like *tiny* and *huge* or colours along the colour spectrum are examples of such symbols. Other AI knowledge could use the PR feature selection techniques intuitively.

Minimum intra-class distance could be applied as a rule to select characteristics of a class that never change, or change only marginally. Maximum inter-class distance could be applied as a rule to select characteristics shared by all classes but of maximum difference in every class. For example, in the horse-donkey example, both horses and donkeys have ears. Horses have small ears and donkeys have large ears. Similarly, brownies, girl guides and nurses all wear uniforms. Brownies have brown uniforms, girl guides, blue, and nurses, white. This would be a good feature for discrimination. The probabilistic techniques to determine independence could be performed on the symbolic knowledge without its transformation to feature space. This would be a quick simple test for redundant knowledge.

Feature selection techniques could also be applied in reverse to generate generalization hierarchies required in many AI applications. A good generalization of several classes would minimize inter-class distance. It could be applied as a rule that searched for a feature that was common among all classes and always had the same value. The leg-count measurement in the horse-donkey problem is an example of minimum inter-class distance; horses and donkeys both have 4 legs.

Feature selection techniques can also be added to an existing AI program as a method of evaluating existing knowledge in a knowledge store. It could also be of use in self-improving systems. This will be discussed in Chapter 6.

5. Problem-Solving Techniques

Having all the knowledge in the world is of no use to a machine intelligence program if there is no means by which to reason with that knowledge. Knowledge, and the existence of a problem-solving and inference method, implicitly or explicitly, are the basis for any machine intelligence program.

5.1. A Comparison of PR and AI

Pattern Recognition and Artificial Intelligence have very different problem-solving techniques. The difference is a reflection of the knowledge representation schemes used by the two fields; consequently, even decision-theoretic PR and syntactic PR differ in their problem-solving approach.

In decision-theoretic PR, problem-solving implies the application of classifiers to patterns described in feature space. In syntactic PR, it suggests the recognition of pattern primitives using decision-theoretic techniques, and the parsing of the resultant pattern to check for structural matching. Problem-solving in the field of AI is distinguished by search, deduction, inference, theorem proving, planning and common sense reasoning [2].

AI uses symbolic reasoning to manipulate the knowledge that it has stored as symbols. Conversely, decision-theoretic PR uses algebraic, statistical and probabilistic techniques, in a mathematical framework, to solve its problems. Syntactic PR may liberally be thought of as a hybrid of the two conceptual approaches; mathematical techniques are used to identify pattern primitives, which are represented as symbols and the parsing of structural knowledge is a symbolic manipulation approach.

Problem-solving techniques exploited by the two fields may be compared at a more abstract level, the mode of reasoning. Decision-theoretic PR uses a *bottom-up* or *data-driven* method of reasoning. It begins with the data, and proceeds to apply the problem-solving reasoning to produce the classification of the pattern. The opposite of bottom-up reasoning is *top-down* or *goal-directed* reasoning. This mode takes the goal and recursively applies problem-solving techniques to create subgoals, until each subgoal is solvable. Humans often apply top-down reasoning in their problem-solving strategy. Similarly, many Artificial Intelligence programs also use top-down reasoning. Unfortunately, top-down reasoning does not always take the given data into consideration when generating possible solutions. Thus it is sometimes inefficient. There is another reasoning method commonly incorporated into

AI programs, called ***means-ends analysis***. Means-ends analysis involves a combination of both top-down and bottom-up reasoning. A difference measure is taken between the current state and the goal state. An operator is prescribed to reduce the difference. If the operator cannot be applied to the goal state, subgoals are set up recursively until an operator can be applied. Means-ends analysis was first applied in Newell's General Problem Solver (GPS) [31]. It was motivated by the idea that humans perform a type of means-ends analysis when solving problems.

Syntactic PR can use a combination of top-down and bottom-up techniques, although not in the same manner as means-ends analysis. The recognition process to identify pattern primitives applies a bottom-up approach. The parsing procedure to detect structural relationships between pattern primitives can either implement a top-down or a bottom-up parse.

PR problem-solving techniques are applicable to solitary problems. Given a problem description, the black box PR system will deliver an answer. PR systems are not appropriate for problems that require man-machine communication. Conversely, AI programs thrive in an interactive problem-solving environment. Knowledge is mappable to real world, user understandable knowledge. Hence, explanations may be generated. The PR classifiers discussed in this paper process feature information in parallel; all knowledge is considered at once. The feature space is cleaved into class spaces, and the unknown pattern is mapped into one of these subareas. Thus, the algorithms yield no room for interaction. AI systems solve problems state-by-state until they reach their goal. At each new state, a new decision must be made. Thus, it seems natural to query the user for data required by the problem-solver, if needed.

The concept of a single stage classifier that utilizes all knowledge at once, in parallel illustrates some limitations of the PR approach. The single stage approach requires that all the classes represented in the classification problem share common, distinguishing features. For example, it would be difficult to put birds into the horse-donkey problem. The height feature would uniquely distinguish them, but birds have no ears with which an ear size measurement could be taken. The problem is more complex in larger domains or where more classes are contained within the classification problem.

There is another disadvantage to the PR approach, if applied to some tasks normally performed by AI programs. This is the requirement that all data must be provided before classification can be performed. This limits the application of single stage PR techniques to problems where the required data is known even if the classification of that data is not. This approach would be unsuitable to many medical diagnosis systems. These systems are often designed as doctor's assistants. The requirement for specific data is related to a partial prognosis of the disease, as a result of initial data manipulation. If the doctor had collected all the data required by the system beforehand, the doctor would quite possibly know what the diagnosis was as well.

To combat the common feature requirement of single stage PR systems, some multi-stage classification schemes have been designed. If there are K distinct classes in the problem, the feature space is not split into K subareas all at once. Areas are split and resplit by the application of different distinguishing features, between different subsets of the K classes. This type of classifier makes decision-making more efficient when there are many classes with many different features, each of which do not discern between all classes.

Single stage PR systems have also been permuted into internally interactive systems. This strategy is commonly exploited on large problems with a great deal of data. A decision-making system can take a pattern descriptor and either perform a classification or make no external decision, and request more information from the internal pattern description. In this way, the decision-making procedure can make a crude classification and then request more specific knowledge to obtain an exact classification. This tactic reduces the dimensionality of the classification task, but it does not address the requirement of entering all possible data into the system originally. Because of the low level of processing of PR systems, it would be more difficult to provide a user interface than in an AI system. Therefore, PR techniques should be applied to problems implementable in a solitary environment, and requiring no interaction or understanding by the user.

An important aspect of Pattern Recognition is its ability to represent and consequently process inexact knowledge. This is a reflection of the probabilistic and statistical basis of PR techniques. The processing of inexact, noisy knowledge is achieved in both decision-theoretic and syntactic PR through the use of measures of similarity. The inability to represent variable knowledge is a weakness in AI systems. However, AI does have means by which ***uncertainty in reasoning*** about knowledge may be performed. Examples of uncertainty in reasoning are found in the expert systems MYCIN [17] and PROSPECTOR [17] and in the application of fuzzy logic [17].

It is important to stress the difference between *variability* in knowledge and *uncertainty* in knowledge. Pattern Recognition represents variability in knowledge by representing a class in terms of its mean and covariance within feature space or by the pdf generated by the training samples. For example, it means that donkeys are not required to have 6 inch ears. Their ear size can vary between 4 inches and 8 inches and still be acceptable. Similarity measures such as MICD and MAP, and error-correcting parsers provide a means of classifying unknown patterns. In contrast, uncertainty in knowledge means that a fact may or may not be true. For example, most cats have long tails, but there are cats that do not. Therefore, the feature *long tails* in classifying cats is uncertain knowledge. In PR, uncertainty is represented and measured by stochastic languages and maximum likelihood parsers. The MAP and MICD classifiers allow for uncertainty in reasoning implicitly because they are measuring similarity.

The AI expert system PROSPECTOR finds ore deposits from geological data. Uncertainty is implemented by the use of Bayes theorem. Prior probabilities of the occurrence of various individual minerals and the probabilities of certain physical characteristics given certain minerals are known. Bayes theorem can compute, how likely a certain mineral is to occur, given collected data. This application of Bayes theorem is almost identical to the very popular MAP classifier of decision-theoretic PR.

The AI expert system MYCIN gives advice on the diagnosis and treatment of infectious blood diseases. It incorporates uncertainty in reasoning in a fashion similar to that of stochastic languages. In the MYCIN system, each IF *condition* THEN *action* rule is assigned a probability. Simple probability rules are applied to select production rules just as stochastic language parsers select a derivation. MYCIN also defines measures of belief and disbelief associated with each rule. These combine to create a *certainty factor* for the hypothesis being tested. PROSPECTOR and MYCIN exemplify the ease with which mathematical measures can be incorporated into an AI system.

5.2. Application of PR Techniques to AI

As mentioned previously, AI techniques have been applied to PR systems for the implementation of high level semantic processing. Limited work has been done in explicitly applying PR techniques to AI systems. There is sometimes a fine line of distinction between PR systems incorporating AI techniques and AI systems incorporating PR techniques. Examples of domains where this is the case are speech and image understanding. Speech and image analysis problems have been addressed by both PR and AI researchers. Speech analysis [20] is speech *recognition* in PR and speech *understanding* in AI, similarly for image analysis. Speech understanding uses syntactic and semantic processing, utilizing conventional speech recognition techniques to analyze primitive sound data. Similarly, there are speech recognition systems that implement high level syntactic and semantic processing using AI techniques.

A good example of speech understanding, and more generally, of the ability of PR problem-solving techniques to be implemented within an AI framework, is HEARSAY [11] [28]. HEARSAY, developed at CMU, was one of several speech understanding projects funded by DARPA in the mid 70's. HEARSAY-II, the successor to HEARSAY-I, had relatively good success as a speech understanding system. However, it is best known for its unique architecture. As opposed to having one central knowledge base and one problem-solving technique, HEARSAY-II had 12 knowledge sources (KSs) each with unique knowledge and a tailored reasoning strategy. The 12 knowledge sources were each experts in a subproblem area of speech understanding. For example, there were syntactic, pragmatic, prosodic, and acoustic KSs etc. The different KSs communicated with each other via a global working memory called a *blackboard*. On the blackboard, information of different type and level was all integrated into a uniform representation scheme. Each KS posted hypotheses to the blackboard, based on information already on the blackboard, and new information yielded by the KS. Specific knowledge probabilities for success were associated with each hypothesis. A controller was used to focus attention on the most likely hypothesis. This unique architecture allowed for low level knowledge sources to be implemented using decision-theoretic PR techniques. The high level KSs applied conventional AI techniques, and all were able to communicate and reason together. The modularity simplified the system.

HEARSAY-II architecture is ideal for integrating Pattern Recognition into any problem domain. The existence of the blackboard eliminates many of the interface worried of implementing the two methodologies together. HEARSAY-II has yielded HEARSAY-III, a software tool and domain-independent AI framework for expert system.

An interesting application of PR problem-solving techniques to a medical expert system, would be to give the system limited sensory perception via PR techniques. For example, imagine a medical diagnosis system that could also listen to a pulse, or hear a murmur, or listening to breathing patterns, while having access to medical knowledge and also patient-specific information. Electro-cardiogram (ECG) [15] [27] or digitized X-rays could be input into the system along with regular medical expert system data. It would yield a more powerful system with not only access to a doctor's mind, but also to his/her eyes and ears.

PR techniques could also be incorporated into an AI framework to perform a crude classification to reduce the dimensionality of the search space. AI techniques could then be applied to fine-tune the decision. Generalized data on the problem that was structured towards use by PR techniques could be quickly processed in parallel, narrowing the problem space immensely. Conversely, decision-theoretic PR could be used in an AI framework similar to its use in syntactic PR. PR could be employed to process low level primitives which could be combined and represented as symbols. These symbols could be manipulated by conventional AI programming techniques.

PR concepts have been applied to game playing. The idea of features has been used to create evaluation functions. These evaluation functions are used as heuristics in a heuristic search of the state space. Originally, Samuel designed evaluation functions for checkers. Typical features included piece advantage, control of the centre etc. The PR problem was to select a weighting for these features. The linear combination of features required to generate the best search had to be learned using an adaptive training mechanism. Rendell [29] [30] did similar work with the 15-puzzle; this will be discussed in the next chapter.

In Chapter 4, some methods for applying PR to the computer representation of knowledge were proposed. Each of these composite representations requires a complimentary problem-solving technique. The procedural representations and production system representations suggested were self-explanatory. They were simply small classifiers inserted into AI programs. The semantic net permutation requires more sophisticated problem-solving methods. It was put forth that semantic nets be represented using formal language grammars. Grammars have been defined for representation of string, trees and graphs in syntactic PR; thus applications are numerous. The major advantage of representing semantic nets as language grammars is to employ the accompanying PR problem-solving techniques. Specifically, error-correcting parsers could be used to handle noisy and distorted data. These error-correcting parsers implement similarity measures for symbolic data. This would be of great advantage in specific situations of AI domains where noisy data is found. General AI techniques in this situation are weak. Another technique for dealing with noisy data is to represent semantic nets using stochastic grammars. Each production in the grammar has associated probabilities. A maximum likelihood parser could be used for reasoning. Finally, the detection of graph isomorphisms has been used to match attributed graphs, another syntactic PR representation. It is debatable whether the time required to detect graph isomorphism would justify their use in an AI environment where required processing capabilities are perhaps less powerful.

Finally, it should be reiterated that many of the decision-theoretic PR classifiers could be adapted to an AI environment with a slight transformation of data or redefinition of similarity. Similarity was previously redefined in the creation of error-correcting parsers. The MAP classifier was the most powerful decision-making technique discussed. Bayes theorem is the foundation for this classifier. It is used very successfully in PROSPECTOR for reasoning with uncertainty. It could be similarly implemented in other domains. Bayes theorem requires a priori knowledge of certain probabilities or estimation of pdfs. If this knowledge is not available, one of the distance metrics could be applied with slightly poorer performance resulting.

6. Learning

One of the most important aspects of human intelligence is the ability to learn. Learning has confounded researchers in many diversified fields for years. Computer scientists have not been immune to this confusion. Machine learning has been a topic of study for several decades; only limited success has been achieved. The motivation for research into learning is two-fold [7]. Learning is studied as a pure science to help psychologist and epistemologists better understand the functioning of the brain and how knowledge relates to that functioning. More practically, with the advent of AI expert systems, has come the requirement for automated knowledge acquisition. Expert systems now require a domain expert. Development costs are high and knowledge acquisition is time consuming. On the other hand, PR systems have needed machine learning since their conception. Because of the problem domains PR systems currently addressed, and the problem-solving methodology which is not suited to human interaction, PR systems always require some degree of machine learning. Machine learning is a large and diverse area of study. This chapter briefly relates work in PR with pertinent AI research.

6.1. A Comparison of PR and AI

Pattern Recognition and Artificial Intelligence have both addressed the problem of machine learning. Many of the techniques originally developed in PR were related to fundamental problem-solving methods. They were not explicitly labelled as machine learning techniques until their operation was analyzed. In contrast, the AI research concerning machine learning has always been identified as such; machine learning is generally recognized as a distinct area of AI. However, machine learning is less crucial to basic AI decision-making algorithms. As a reflection of the differences in methodology, PR and AI have somewhat different approaches to machine learning. Pattern Recognition has used mathematical techniques to solve problems in machine learning; symbolic learning methods have been implemented in AI. However, there have been exceptions to this rule. Several AI researchers such as Samuel [8] and Rendell [29] [30] have applied mathematical learning techniques, of a PR nature, to problems of heuristic search in game-playing.

There are several different types of AI machine learning: *rote learning*, *learning by being told*, *discovery learning*, *learning by analogy* and *learning by example.* Pattern Recognition addresses only two, rote learning and learning by example.

Rote learning involves the memorization or recording of data. In a sense, all computer programs are capable of rote learning. In PR systems, rote learning is performed by the K-nearest-neighbours classifier, where training samples, characterizing the boundaries of the class, are learned and stored without any transformations being performed. Learning by example, or *induction*, is the common type of learning performed by PR systems. In learning by example, a system is given examples of how it should perform. The systems must group together this information and make generalized rules that can be applied to new input data. This is exactly what all PR systems do. They take training samples and generate a classifier for use on subsequent unlabelled data. Therefore all PR systems are instances of learning by example.

To complete the descriptions, learning by being told is comparable to a system taking advice. It involves accepting high-level knowledge and transforming it into a form that is usable by the system. On the other hand, in a discovery learning system, a set of elementary knowledge primitives are given and it must apply these to discover new concepts. Finally, learning by analogy is a difficult form of machine learning. It involves taking knowledge from another domain, and identifying analogous relations with knowledge in the problem-domain. The relevant knowledge may then be transferred and applied to the problem-domain.

Machine learning exists in at least three different forms: as a knowledge acquisition tool to develop initial problem-solving rules, as a mechanism to improve system performance, and as a complete machine learning program whose purpose is to acquire new knowledge by one or more learning techniques. Artificial Intelligence has implemented symbolic machine learning in all three forms. Pattern Recognition has accomplished machine learning in the first and the third.

Every PR system must automatically acquire knowledge to develop problem-solving rules. In a decision-theoretic system, it involves the selection of features from training samples and the subsequent construction of a classifier. AQ11 and META-DENDRAL are examples of the few AI applications to actually perform initial knowledge acquisition to generate rules for a task; they are both instances of learning by example.

Self-improving systems are not common in AI. Samuel's checker program is an instance of an adaptive AI learning program [8]. Samuel tried three different approaches to get a computer to learn to play checkers. One of these approaches involved building a heuristic evaluation function to optimize the search for a solution. Features of the game were selected. The learning problem was to create a linear function weighting each feature appropriately to yield an optimal solution. An adaptive training method was used to calculate the weights. After each program run, the weights were updated until a satisfactory heuristic was obtained. Doran, Michie and Rendell have done similar work and extensions with the 8-puzzle and the 15-puzzle. This adaptive learning mechanism is a direct application of PR to AI machine learning.

TEIRESIAS [17], a tool for the acquisition of knowledge in the expert system MYCIN. If MYCIN fails, TEIRESIAS is capable of both suggesting what kind of rule will correct the problem and of writing a form of the rule. TEIRESIAS interacts with a human when performing such improvements. Other expert systems tools exist for similar knowledge-base revisions [17].

In PR, adaptive or training techniques are sometimes used to learn the classifier. The classifiers are self-improving in the sense that they "train" until satisfactory performance is obtained. After which, the classifier is static. The self-improving nature of the system only exists during the analysis phase or during updating; self-improvement is not integrated into the recognition stage. Thus there is no ongoing mechanism for improving system performance.

Finally, most of the existing AI learning programs have learning as their primary task. Examples of such systems are Lenat's AM [8], Winston's work on learning structural descriptions from examples [6] and Mitchell's LEX [8]. PR clustering programs are examples of PR systems whose primary task is to learn. They find naturally occurring clusters in samples of unlabelled data. Clustering techniques exist for both decision-theoretic and syntactic PR. Grammatical inferencing programs ar also an example of syntactic PR programs designed solely to learn.

Machine learning techniques like all other problems in machine intelligence, are very much a function of the utilized knowledge representation scheme. AI knowledge representation addresses a much larger problem domain than PR. As a result, AI has attempted to tackle many different types of learning problems using at least five different learning methods. A criticism of most AI learning programs is that they are not self-sufficient; they require domain specific knowledge to assist in the learning procedure. In contrast, PR learning methods require no domain specific knowledge. Mathematics is a very powerful framework within which to operate, as illustrated by the success of PR in machine learning. Because the various PR learning techniques are not domain specific, they are well suited as learning tools. These tools could be "plugged in" to an AI program. AI learning techniques appear to be less adaptable. Like most AI programs, they ar fine-tuned to the specific domain.

6.2. Application of PR Techniques to AI

The greatest number of applications of PR techniques to AI have been in the area of machine learning. PR has exhibited superior learning capabilities within its methodology. Some PR learning tools have been applied, as is, to the AI environment. Others have been adapted to perform the same conceptual task using symbolic reasoning. There is potential for much more interaction. PR has three distinct techniques that could act as tools in an AI framework: knowledge selection, adaptable classifiers, and independent learning tools such as clustering and grammatical inference.

Knowledge selection was described in detail in Chapter 4. It has many conceivable applications to AI in the area of original knowledge acquisition. Knowledge selection could also be applied to evaluate the knowledge being employed for decision-making, once the system is in place. Many AI systems do not have access to training data at their conception. Therefore, it is more difficult to apply knowledge selection techniques initially. However, once an AI system is operational and generating results, knowledge selection could be used as a learning procedure to check the efficiency of the knowledge being employed to generate decisions. Knowledge selection could improve the quality of the knowledge by detecting redundancies and inapplicable data, and by proposing a minimum basis to represent the required knowledge.

A limitation of many of the AI learning programs is that they require domain specific knowledge. Mitchell's LEX is an example of such a system. LEX learns to solve elementary problems of symbolic integration. It requires a generalization hierarchy of concepts before it can be implemented. This hierarchy must be created manually. Knowledge selection techniques could be implemented to assist in building and streamlining such a hierarchy. All nodes at a particular level of the hierarchy would be independent or distinct. A group of independent nodes would be attached to a common ancestor. The ancestor would be the conceptual grouping of all the common features of the independent nodes. They would be characterized as all the features that were not discerning in the creation of a classifier.

The concept of an adaptive classifier has been applied to AI for some time. Rendell [29] [30] proposed its use for state-space learning directed to games such as the 15-puzzle. Rendell's work is a good example of the power of PR methodology applied to a symbol system design. The original probabilistic learning system, PLS1, is similar in nature to Samuel's system previously described. Features are combined to form an evaluation function. The evaluation function assists in performing a heuristic state-space search. Adaptive methods are utilized to select weightings for the features. Statistical performance measures from solutions to the 15-puzzle help to generate the probability of usefulness of a task. PLS1 generates locally optimal feature weights. Rendell handles the issue of noisy data via probabilistic measures and clustering. PLS2 is an extension of PLS1 to accommodate feature interaction.

Adaptive learning techniques could be applied in other areas of AI. The trainable statistical classifier was employed to develop approximation of the Bayes classifier. PROSPECTOR is an example of an expert system that utilizes Bayes theorem for reasoning with uncertainty. It could implement an adaptive statistical approach to Bayes theorem. This would ensure optimal probabilities.

An interesting technique that might be applied to AI in a state-space learning problem is the learning of production probabilities in a stochastic grammar [34]. This is an inductive learning technique that estimates the production probabilities. It would be particularly suitable for applying reasoning with uncertainty to a semantic net or graphical AI KR scheme. This would aid search and make the system more efficient. The technique requires a set of samples, to be implemented.

Of the independent PR tools both grammatical inference and clustering are applicable to machine learning. Grammatical inference is an instance of symbolic learning by example and as such it is directly related to Winston's work on learning structural descriptions from examples. There are several different methods applied to the problem of grammatical inference. One of these methods involves the use of positive and negative examples and the application of generalization and specialization; these techniques are reminiscent of AI symbolic learning. Grammatical inference has also used statistical approaches to generate a structural hierarchy. Bayes theorem is employed to help select the most probable grammar from training instances. There are several other enumerative and constructive techniques [8] that could be applicable to AI learning research beyond the scope of Winston's work. Grammatical inference is a very specific learning problem. Unless some of the formal language representations are implemented in an AI framework, grammatical inference will only be applicable as an exercise in AI machine learning.

Clustering is another technique commonly utilized but not originally conceived in PR. It has its foundations in numerical taxonomy. In recent years, several AI researchers have implemented clustering in inductive learning systems. Michalski [21] [22] [23] and Langley [19] have been foremost among these researchers. As with many other AI researchers, they have been critical of the lack of semantic processing in the very mathematical methods used to implement clustering. The common complaint is that the approach does not allow for symbolic knowledge and that the clusters generated have no intuitive meaning. This criticism may be valid for specific AI learning applications but it does not warrant abandoning numerical clustering. There are still many areas of AI where numerical clustering techniques may be applied.

Both Michalski and Langley have implemented *conceptual clustering*, clustering without numerical tools. Conceptual clustering could be described as an intuitive approach to clustering. Michalski has had a great deal of success with his CLUSTER/2 program which employs a technique comparable to a conceptual K-means algorithm. Objects are represented as a series of conjunctive concepts in VL_1, the variable-valued logic described in Chapter 4. As a result, both numeric and symbolic data may be represented in a symbolic framework. Using a set of positive and negative examples of concepts, CLUSTER/2 selects K seeds. Each seed is used as a prototype for one of the K classes. The other K-1 seeds are negative instances of that class. By clustering the remaining data, a set of descriptions is generated, covering each seed. From these clusters, new seeds are generated and the algorithm repeated until the seeds become stable. The resultant seeds each represent one of K classes. To generate a hierarchical description of each class, the CLUSTER/2 procedure is applied recursively, generating nodes and branches for the class.

Langley takes a slightly different approach to conceptual clustering. Based on some of the work by Quinlan [26], Langley's DISCON system generates discrimination networks. Instead of using positive and negative instances, Langley considers all observed data to be positive examples, and all unobserved data to be negative examples. From the discrimination network, a classification tree is generated.

Both of these methods have utilized the concept of clustering in a symbolic framework. Conceptual clustering is a good technique for learning by example. It is an application of PR in an AI framework. Unfortunately, it is not very good at dealing with complex problems with variability in data measures [10]. Numerical clustering could also be applied to the problem. It holds many of the advantages of PR previously stated. In particular, a class is often represented by many slightly different objects and numerical clustering uses mathematical measures of similarity to overcome this common problem. Much of the symbolic data in an AI program could be mapped into a mathematical framework to take advantage of numeric clustering techniques. There are also clustering techniques for dealing with some nominal data [35]. These remarks are not in criticism of the conceptual clustering approach, but a remark in support of the application of other types of clustering in an AI environment.

Database management is another field of computer science having application to AI. Many of the problems of managing large stores of knowledge are common to both fields. As knowledge on various problem domains increases in size, it becomes more difficult for domain experts to see all the relationships between knowledge. The medical profession is a good example of such a domain. Recently, work has been done on the creation of an intelligent database management system (dbms) [32] that finds relationships between attributes stored in the database using conventional clustering techniques. At the moment it only deals with mathematical data, but the possibilities are numerous. Imagine such clustering techniques being applied to a knowledge-base. Numerical clustering techniques, statistical methods, or even conceptual clustering applied to knowledge could result in the finding of relationships not even known by the domain experts. The advantage the system has over the human expert is that it can consider many different attributes at the same time and perform as many statistical comparisons as desired. This technique could also be applied to a medical database of patient histories. If the system were allowed to identify and cluster attribute relations it might discover subtle hereditary or lifestyle relationships between diseases.

This technique also has implications to the creation of self-improving systems, where initial knowledge selection is difficult. If a cluster was made of each distinct class or diagnosis and the results from the decision-making tests grouped together, it might be possible to identify tests that predicted the correct class. Tests yielding incorrect or irregular results could be omitted.

Finally, syntactic PR has applications to the AI study or learning by analogy. Although the syntactic PR KR scheme is limited in applicability, it is able to yield a hierarchical representation of the blocks world. The blocks world is frequently used as a simple domain upon which to test theories of learning. In learning by analogy knowledge entities are not matched so much as the relationships linking those knowledge entities. Pattern grammars are a means of representing those structural relationships.

To test to see whether a foreign pattern is analogous to a pattern of a pattern grammar, the foreign pattern could attempt to be parsed by the pattern grammar. Patterns and subpatterns in the grammar could be represented as variables. The variables could be maintained to ensure consistency in the patterns and subpatterns but not necessarily a match between the specific values of the foreign pattern and the pattern grammar. The structural relationships are more important. If a foreign pattern did not parse using the start goal, then the subgoals could be replaced as start goals and parsing attempted again. The parsing technique would also yield a description of where the foreign pattern did not match the grammar. The description of a pattern as a grammar allows for the representation of specific rules recursively where multiplicity exists. The matching of multiplicity is one of several problems to be addressed in learning by analogy.

7. Conclusions

Pattern Recognition and Artificial Intelligence differ a great deal in methodology. AI attempts to validate Newell's *physical symbol system hypothesis* by representing knowledge as symbols and symbol structures. A mapping of knowledge to the real world is retained. A separate set of processes are defined for symbolic manipulation of this knowledge. In contrast, PR takes a nonsymbolic approach to the problem of machine intelligence. Knowledge is reduced to concise mathematical representations. No simple mapping to the real world exists. Algebraic and multivariate statistical techniques are used for manipulation of this knowledge; knowledge and the reasoning mechanism are intertwined. Syntactic PR is a step towards the symbol system approach. A hierarchical symbol system structure is placed on top of the mathematical framework, for processing of structural knowledge. The difference in methodology provides the basis for the contrast in techniques applied to issues of knowledge representation, problem-solving and learning.

PR techniques are very powerful. However, they are not capable of solving all the problems of machine intelligence. Throughout the paper, specific areas where PR tools are applicable to problems in AI have been briefly highlighted. PR is a vast field. Relevant techniques are not restricted to those mentioned herein. There is potential for much interaction. It is hoped that this paper has provided the stimulus for further research into the utilization of PR methods as tools of AI machine intelligence.

References

[1] Banerji, **Theory of Problem Solving**, American Elsevier Pub Co, New York, 1969

[2] Barr, Feigenbaum, **The Handbook of AI, Vol I & II**, William Kaufmann Inc, California, 1981

[3] Batchelor, Beck, *"Diagnosis and Data Structure Analysis of Migraine and Headache"*, Pattern Recognition Proc. 1976

[4] Batchelor (editor), **Pattern Recognition Ideas in Practice**, Plenum Press, New York, 1978

[5] Boyle, *"The Decision to Grant Credit"*, PhD Thesis, MIT, 1974

[6] Carbonell, Michalski, Mitchell, **Machine Learning: An Artificial Intelligence Approach**, Tioga Press, California, 1983

[7] Cohen, *"A Powerful and Efficient Structural Pattern Recognition System"*, Artificial Intelligence, Vol 9, December 1979

[8] Cohen, Feigenbaum, **The Handbook of AI, Vol III**, William Kaufmann Inc, California, 1982

[9] Duda, Hart, **Pattern Classification and Scene Analysis**, John Wiley & Sons, 1973

[10] Dale,M.B., *"On the Comparison of Conceptual and Numerical Taxonomy"*, IEEE Trans on Pattern Analysis and Machine Intelligence, Vol PAMI-7, No. 2, March 1985

[11] Erman, Hayes-Roth, Lesser, Reddy, *"The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty"*, Computing Surveys 12, 1980

[12] Fu (editor), **Digital Pattern Recognition**, Springer-Verlag, New York, 1976

[13] Fu (editor), **Syntactic Pattern Recognition Applications**, Springer-Verlag, New York, 1977

[14] Fu, **Syntactic Pattern Recognition and Applications**, Prentice-Hall, New Jersey, 1982

[15] Fu, **Applications of Pattern Recognition**, CRC Press, Florida, 1982

[16] Fu, *"A Step Towards Unification of Syntactic and Statistical Pattern Recognition"*, IEEE Trans on Pattern Analysis and Machine Intelligence, Vol PAMI-5, no. 2, March 1983

[17] Hayes-Roth et al (editors), **Building Expert Systems**, Addison-Wesley, Don Mills, 1983

[18] Jernigan, *"Pattern Recognition Course Notes"*, University of Waterloo, March 1979

[19] Langley, *"Conceptual Clustering as Discrimination Learning"*, Proc. CSCSI, 1984

[20] Levinson, Liberman, *"Speech Recognition by Computer"*, Scientific American, Vol 244, no. 4, April 1981

[21] Michalski, *"Pattern Recognition as Rule-Guided Inductive Inference"*, IEEE Trans on Pattern Analysis and Machine Intelligence, Vol PAMI-2, no.4, July 1980

[22] Michalski, Stepp, *"Revealing conceptual structure in data by inductive inference"*, in **Machine Intelligence 10**, Hayes et al editors, Ellis Horwood, Chichester, 1981

[23] Michalski, Stepp, *"Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy"*, IEEE Trans on Pattern Analysis and Machine Intelligence, Vol PAMI-5, No. 4, July 1983

[24] Morris (editor), **The Houghton Mifflin Canadian Dictionary of the English Language**, Houghton Mifflin Canada Ltd., Markham, 1980

[25] Pao, Ernst (editors), **Tutorial: Context-Directed Pattern Recognition and Machine Intelligence Techniques for Infromation Processing**, IEEE Computer Society Press, California, 1982

[26] Quinlan, *Semi-autonomous acquisition of Pattern-base Knowledge* in Hayes et al (editors) *Machine Intelligence 10*, Ellis Horwood, Chichester, 1981

[27] Raeside, *"An Application of Pattern Recognition to Echocardiography"*, IEEE Trans. on Systems, Man and Cybernetics, Vol SMC-8, Feb. 1978

[28] Reddy, Erman, Fennell, Neely, *"The Hearsay Speech Understanding System: An example of the Recognition Process"*, 3rd Int. Joint Conf on AI, 1983

[29] Rendell, *"Towards a Unified Approach for Conceptual Knowledge Acquistion"*, The AI Magazine, Winter 1983

[30] Rendell, *"A New Basis for State-Space Learning Systems and a Successful Implementation"*, Artificial Intelligence 20, 1983

[31] Rich, **Artificial Intelligence**, McGraw-Hill, Toronto, 1976

[32] Shen, Kamel, Wong, *"Intelligent Data Base Management Systems"*, Proc. Int. Conf. on Systems, Man and Cybernetics, 1983

[33] Sykes,J.B (editor), **The Concise Oxford Dictionary**, Sixth Edition, Oxford University Press, 1976

[34] Tou, Gonzalez, **Pattern Recognition Principles**, Addison-Wesley, Don Mills, 1974

[35] Watanabe, **Knowing and Guessing, A Formal and Quantitative Study**, John Wiley & Sons Inc, Toronto, 1969

[36] Wong, Lu, *"Representation of 3-D Objects by Attributed Hypergraphs for Computer Vision"*, Proc of Int Conf on Systems, Man and Cybernetics, 1983

[37] Wong, Wou, *"Entropy and Distance Measures of Random Graphs"*, IEEE Computer Vision and Pattern Recognition, 1983

Supplementary Readings

Pattern Recognition

- Batchelor (editor), **Pattern Recognition Ideas in Practice**, Plenum Press, New York, 1978
- Duda, Hart, **Pattern Classification and Scene Analysis**, John Wiley & Sons, 1973
- Fu (editor), **Digital Pattern Recognition**, Springer-Verlag, New York, 1976
- Fu (editor), **Syntactic Pattern Recognition Applications**, Springer-Verlag, New York, 1977
- Fu, **Syntactic Pattern Recognition and Applications**, Prentice-Hall, N.J., 1982
- Fu, **Applications of Pattern Recognition**, CRC Press, Florida, 1982.
- Tou, Gonzalez, **Pattern Recognition Principles**, Addison-Wesley, Don Mills, 1974
- Watanabe, **Knowing and Guessing, A Formal and Quantitative Study**, John Wiley & Sons Inc, Toronto, 1969

Artificial Intelligence

- Barr, Feigenbaum, **The Handbook of AI, Vol I & II**, William Kaufmann Inc, California, 1981
- Cohen, Feigenbaum, **The Handbook of AI, Vol III**, William Kaufmann Inc, California, 1982 1973
- Hayes-Roth et al (editors), **Building Expert Systems**, *Addison-Wesley, Don Mills, 1983*
- Rich, **Artificial Intelligence**, McGraw-Hill, Toronto, 1976
- Winston, **Artificial Intelligence**, 2nd Edition, Addison-Wesley, 1984