Unit 1 Introduction to Pattern Recognition

Structure:

1.1 Introduction

Object recognition is a task performed daily by living beings and is inherent to their ability and necessity to deal with the environment. It is performed in the most varied circumstances with remarkable efficiency. Recognizing objects is considered here in a broad cognitive sense may consist of simple to complex tasks. The development of methods capable of emulating the most varied forms of object recognition has evolved along with the need for building "intelligent" automated systems, the main trend of today's technology in industry and in other fields of activity as well. In these systems objects are represented in a suitable way for the type of processing they are subject to. Such representations are called patterns.

Pattern Recognition (PR) is the scientific discipline dealing with methods for object description and classification. Pattern Recognition is a fertile area of research, with multiple links to many other disciplines, involving professionals from several areas.

In this unit, we would be discussing the fundamental concepts of Pattern recognition systems, the basic terminology associated, the machine's perception of patterns, and the design cycle of a pattern recognition system.

As a prerequisite knowledge, you need to know basic calculus, elementary linear algebra, and basics of probability theory.

Objectives:

After studying this unit, you should be able to:

- explain various terms associated with Pattern Recognition
- describe how a machine interprets the input patterns
- explain the components of a Pattern Recognition system
- illustrate the design cycle of a Pattern Recognition system
- explain the types of pattern learning methods

1.2 Terminology in Pattern Recognition

In this section you would be familiarized with various terms needed to understand Pattern Recognition Systems.

- **Features:** A set of variables believed to carry discriminating and characterizing information about an object, which are usually measurements or observations about the object.
- **Collection:** A collection of d such features, ordered in some meaningful way into d- dimensional column vector is the feature vector, denoted x, which represents the signature of the object to be identified. The d-dimensional space in which the feature vector lies is referred to as the feature space. A d-dimensional vector in a d-dimensional space constitutes a point in that space.
- **Class or Label:** The category to which a given object belongs is called the class (or label) and is typically denoted by o.
- **Pattern:** A collection of features of an object under consideration, along with the correct class information for that object.
- **Instance or Exemplar:** Any given sample pattern of an object is also referred to as an instance or an exemplar.
- **Pattern Recognition System:** The goal of a pattern recognition system is to estimate the correct label corresponding to a given feature vector based on some prior knowledge obtained through training.

- **Training:** The procedure by which the pattern recognition system learns the mapping relationship between feature vectors and their corresponding labels. This relationship forms the decision boundary in the d-dimensional feature space that separates patterns of different classes from each other.
- Therefore, we can equivalently state that the goal of a pattern recognition algorithm is to determine these decision boundaries, which are, in general, nonlinear functions. Consequently, pattern recognition can also be cast as a function approximation problem.
- **Training Set:** The set of samples used in system design.
- **Test Set:** The data set provided for testing the pattern recognition system.
- **Histogram:** A convenient way to describe the data. To form a histogram, the data from a single class are grouped into intervals. Over each interval, a vertical rectangle is drawn, with its area proportional to the number of data points falling into that interval.
- **Feature Vector:** A "Feature Vector" corresponds to each sample, which determines its position in the plot. The feature space can be divided into two decision regions by a straight line called "Linear Decision Boundary". This can be used to classify a sample of unknown class.

  If a feature space cannot be perfectly separated by a straight line, a more complex boundary might be used.

  Alternatively, a simple decision boundary such as a straight line might be used even if it did not perfectly separate the classes, provided that the error rates were acceptably low.

  A very complicated decision boundary could probably separate the two classes in the training data set perfectly, but this overfitting would probably result in poor testing set performance.
- **Digital Image:** A matrix where each number represents the brightness at regularly spaced points or very small regions in the image. These points are called "pixels" (picture elements) and the brightness value of a pixel is called its "Gray Level".
- **Scanners:** These are the devices to convert photographs to digital images.

- **Segmentation:** The first step in building an automatic classification system, which separates objects (particles) from the background.

  After locating the objects, we must extract features from the objects that can be used to identify them.

  The figure 1.1 illustrates these concepts on a hypothetical 2D, four class problem. For example, feature 1 may be systolic blood pressure measurement and feature 2 may be the weight of a patient, obtained from a cohort of elderly individuals over 60 years of age. Different classes may then indicate number of heart attacks suffered within the last 5-year period, such as none, one, two, or more than two.

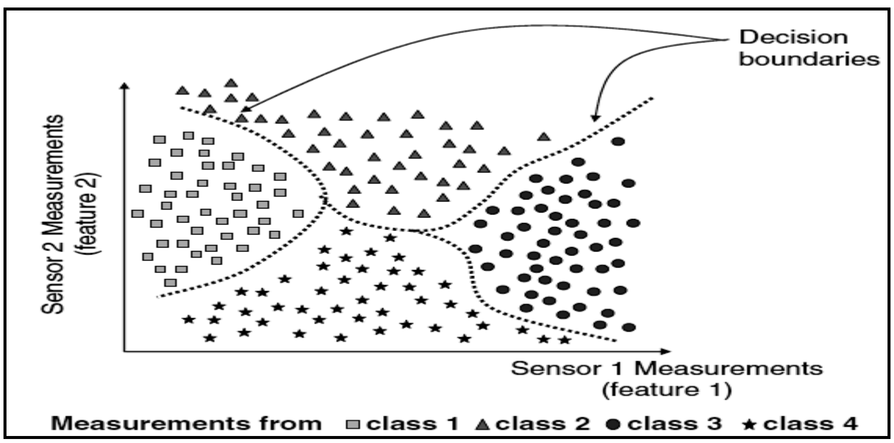

Fig. 1.1: Graphical representation of data and decision boundaries.

The pattern recognition algorithm is usually trained using training data, for which the correct labels for each of the instances that makes up the data is a priori known. The performance of this algorithm is then evaluated on a separate test or validation data, typically collected at the same time or carved of the existing training data, for which the correct labels are also a priori known. Unknown data to be classified, for which the pattern recognition algorithm is trained, is then referred to as field data. The correct class labels for these data are obviously not known a priori, and it is the classifier’s job to estimate the correct labels.

A quantitative measure that represents the cost of making a classification error is called the cost function. The pattern recognition algorithm is specifically trained to minimize this function.

A typical cost function is the mean square error between numerically encoded values of actual and predicted labels. A pattern recognition system that adjusts its parameters to find the correct decision boundaries, through a learning algorithm using a training dataset, such that a cost function is minimized, is usually referred to as the classifier or more formally as the model.

Incorrect labeling of the data by the classifier is an error and the cost of making a decision, in particular an incorrect one, is called the cost of error. We should quickly note that not all errors are equally costly. For example, consider the problem of estimating whether a patient is likely to experience myocardial infarction by analyzing a set of features obtained from the patient's recent ECG. Two possible types of error exist.

The patient is in fact healthy but the classifier predicts that s/he is likely to have a myocardial infarction is known as a false positive (false alarm error), and typically referred to as type I error. The cost of making a type I error might be the side effects and the cost of administering certain drugs that are in fact not needed.

Conversely, failing to recognize the warning signs of the ECG and declaring the patient as perfectly healthy is a false negative, also known as the type II error. The cost of making this type of error may include death. In this case, a type II error is costlier than a type I error. Pattern recognition algorithms can often be fine-tuned to minimize one type of error at the cost of increasing the other type.

- **Training performance:** Two parameters are often used in evaluating the performance of a trained system. The ability or the performance of the classifier in correctly identifying the classes of the training data, data that it has already seen, is called the training performance.

  The training performance is typically used to determine how long the training should continue, or how well the training data have been learned. The training performance is usually not a good indicator of the more meaningful *generalization performance*, which is the ability or the performance of the classifier in identifying the classes of previously unseen patterns.

1.3 Machine Perception

We should try to design and build machines that can recognize patterns. From automated speech recognition, fingerprint identification, optical character recognition, DNA sequence identification, and so on, it clear that reliable, accurate pattern recognition by machine would be immensely useful.

Moreover, in solving the innumerable problems required to build such systems, we gain deeper understanding and appreciation for pattern recognition systems in the natural world – most particularly in humans.

For some problems, such as speech and visual recognition, our design efforts may in fact be influenced by knowledge of how these are solved in nature, both in the algorithms we employ and in the design of special-purpose hardware.

Example: A fish packing plant wants to automate the process of sorting incoming fish on a conveyor belt according to species. As a pilot project it was decided to try to separate sea bass from salmon using optical sensing. Assume that we have set up a camera, took some sample images, and have begun to take a note of some physical differences between the two types of fish – length, precision, width, number and shape of fins, position of the mouth and so on – and these suggest the features to explore for use in our classifier.

We also notice noise or variations in the images – variations in lighting, position of the fish on the conveyor, even "static" due to the electronics of the camera itself.

Given that there truly are differences between the population of sea bass and that of salmon, we view them as having different models – different descriptions, which are typically mathematical in form.

The overarching (or overhead) goal and approach in pattern classification is to hypothesize the class of these models, process the sensed data to eliminate noise, and for any sensed pattern choose the model that corresponds best.

Our proposed prototype system would perform this task taking the following form:

As a first step in this direction, the camera captures the image of the fish. Next the camera's signals are preprocessed to simplify subsequent operations without losing relevant information. In particular, our system may use the *segmentation* operation in which the images of different fish are somehow isolated from one another and from the background. The information from a single fish is sent to a *feature extractor*, whose purpose is to reduce the data by measuring certain features or properties.

These features are then passed to a classifier that evaluates the evidence presented and makes a final decision as to the species.

Self Assessment Questions

1. The ____________________ are a set of variables believed to carry discriminating and characterizing information about an object, which are usually measurements or observations about the object.
2. The procedure by which the pattern recognition system learns the mapping relationship between feature vectors and their corresponding labels is known as _______________.
3. A ______________________ corresponds to each sample, which determines its position in the plot.
4. The operation in which the images of different samples are somehow isolated from one another and from the background is known as _____________.
5. A quantitative measure that represents the cost of making a classification error is called the ____________________.

1.4 Pattern Recognition Systems

Pattern Recognition is the act of taking raw data and making an action based on the 'category' of the pattern.

In pattern recognition, classes or categories indicate the groupings into which the samples are pushed according to the measurements or criteria used.

The following are the two key approaches to pattern recognition:

- Classification – we already know the classes. This indicates that the classes have been labeled or determined prior to classification.

- Clustering – we don't know the kinds of classes. This indicates that the classes would be determined post classification process.

General Definition: "Given some measurements of an unknown pattern, find a useful representation and then assign it to a class or category based on distinguishing properties".

The figure 1.2 shows the basic steps involved in Pattern Recognition:

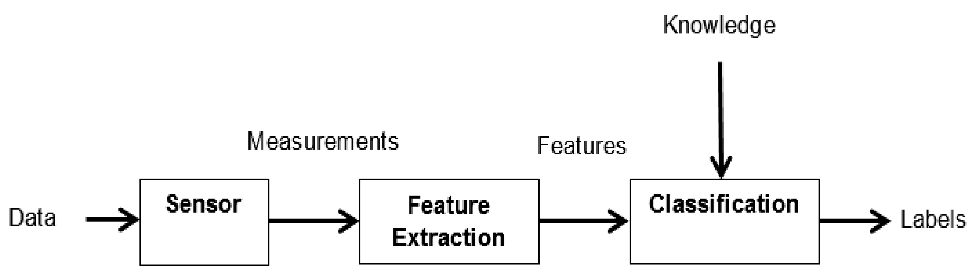

Fig. 1.2: Steps in Pattern Recognition (PR)

Pattern Recognition (PR) is a tool for machine intelligence problem. PR is a science for description or classification/recognition of measurements. Now let us discuss the elements involved in recognizing a pattern of input data.

Data: The input data to the Sensor, which may be in the form of an image, text, and so on.

Sensor: Any device capable of capturing images or any kind of data for processing.

Measurements: The criteria used to classify the input data samples into various classes or categories; Example: The length of a fish to determine its type.

Feature Extraction: A processing module which extracts the features of the samples based on the measurements determined.

Features: The output of the feature extraction module, which extracts predetermined features for further classification.

Knowledge: A database indicating the features to be used for classification depending on the application and input data collected.

Classification: It is the process of putting the sample into a particular class based on the knowledge of extracted features.

Labels: It is the next step to classification wherein the classes are given names or labels like class1, class2,…

Pattern Recognition is a subject researching object description and classification method. It is also a collection of mathematical, statistical, heuristic and inductive techniques of fundamental role in executing the tasks like human being on computers. In a sense, Pattern Recognition is figuring out actual problems via mathematical methods.

The primary goal of pattern recognition is supervised or unsupervised classification. Among the various frameworks in which pattern recognition has been traditionally formulated, the statistical approach has been most intensively studied and used in practice. More recently, neural network techniques and methods imported from statistical learning theory have been receiving increasing attention.

In our previous example of hypothetical fish classification system, we made the distinction between three different operations – preprocessing, feature extraction, and classification. Figure below shows a slightly more elaborate diagram of the components of a typical pattern recognition system. To understand the problem of designing such a system, we must recognize the problems that each of these components must solve.

The following are the interrelated approaches to Pattern Recognition:

- **Statistical Pattern Recognition (StatPR):** It is a classical method of Pattern Recognition based on feature vector distributing which can be derived from probability and statistical models. The statistical model is defined by a family of class-conditional probability density functions $Pr(x|c_i)$ (Probability of feature vector x given class c_i). In StatPR, we put the features in some optional order, and then we can regard the set of features as a feature vector. Also statistical pattern recognition deals with features only without considering the relations between features.
- **Syntactical or structural Pattern Recognition (SyntPR):** The key idea in structural and syntactic pattern recognition is the representation of patterns by means of symbolic data structures such as strings, trees, and graphs. In order to recognize an unknown pattern, its symbolic representation is compared with a number of prototypes stored in a database. In this project, we aim at developing new symbolic matching and parsing algorithms for a variety of applications.

- **Neural Pattern Recognition (NeurPR):** It is a data clustering method based on distance measurement; also this method is model-irrespective. The neural approach applies biological concepts to machines to recognize patterns.

We would discuss the operations of each component and try to understand the possible problems that could arise.

1.4.1 Sensing

This is the action performed by a Sensor. The input to a pattern recognition system is often some kind of transducer, such as a camera or a microphone array. The difficulty of the problem may well depend on the characteristics and limitations of the transducer – its bandwidth, resolution, sensitivity, distortion, signal-to-noise ratio, latency etc.

1.4.2 Feature extraction

Feature Selection is the process of identifying the most effective subset of the original features to use in clustering. *Feature Extraction* is the use of one or more transformations of the input features to produce new salient features. Either or both of these techniques can be used to obtain an appropriate set of features to use in clustering.

Feature extraction techniques try to extract statistical regularities (or sometimes irregularities) directly from the inputs.

An ideal feature extractor would yield a representation that makes the job of a classifier trivial; conversely an omnipotent classifier would not need the help of a sophisticated feature extractor.

The traditional goal of the feature extractor is to characterize an object to be recognized by measurements whose values are very similar for objects in the same category, and very different for objects in different categories.

This leads to the idea of seeking distinguishing features that are invariant to irrelevant transformations of the input. In our fish example, the absolute location of a fish on the conveyor belt is irrelevant to the category, and thus our representation should be insensitive to the absolute location of the fish. Ideally, in this case we want the features to be invariant to translation, whether horizontal or vertical. In general, features that describe properties such as shape, color and many kinds of texture are invariant to translation, rotation and scale.

The problem of finding rotation invariant features from an overhead image of a fish on a conveyor belt is simplified by the fact that the fish is likely to be lying flat, and the axis of rotation is always parallel to the camera's line of sight.

A more general invariance would be for rotations about an arbitrary line in three dimensions. The image of even such a simple object as a coffee cup undergoes radical variation as the cup is rotated to an arbitrary angle: The handle may be occluded – that is, hidden by another part. The bottom of the inside volume come into view, the circular lip appear oval or a straight line or even obscured, and so forth. Furthermore if the distance between the cup and the camera can change, the image is subject to projective distortion. How might we ensure that the features are invariant to such complex transformations? Or should we define different subcategories for the image of a cup and achieve the rotation invariance at a higher level of processing?

In speech recognition, we want features that are invariant to translations in time and to changes in the overall amplitude. We may also want features that are insensitive to the duration of the word i.e. invariant to the rate at which the pattern evolves. Rate variation is serious problem in speech recognition. Not only do different people talk at different rates, but a single talker may vary in rates, causing the speech signal to change in complex ways. How can we make a recognizer that changes its representations for some categories *differently* from that for others under such rate variations?

A large number of highly complex transformations arise in pattern recognition, and many are domain specific. Far more severe are transformations such as *non-rigid deformations* that arise in three dimensional object recognition, such as the radical variation in the image of your hand as you grasp an object or snap your fingers. Similarly, variations in the illumination or the complex effects of cast shadows may need to be taken into account.

As with segmentation, the task of feature extraction is much more problematic, and domain dependent than is classification proper, and thus requires knowledge of the domain. A good feature extractor for sorting a fish would probably be of little use for identifying fingerprints, or classifying photomicrographs of blood cells.

However, some of the principles of pattern classification can be used in the design of the feature extractor. In some cases, they can also be used to select the most valuable features from a larger set of candidate features.

In our fish example, we exactly assumed that each fish was isolated, separate from others on the conveyor belt and could easily be distinguished from the conveyor belt.

In practice, the fish would often be adjacent to or overlapping, and our system would have to determine where one fish ends and the next begins – the individual patterns have to be segmented. If we have already recognized the fish then it would be easier to segment their images.

But how can we segment the images before they have been categorized, or categorize them before they have been segmented? It seems we need a way to know when we have switched from one model to another, or to know when we just have background or “no category.” How can this be done?

Segmentation is one of the deepest problems in pattern recognition. In automated speech recognition, we might seek to recognize the individual sounds (e.g. phonemes, such as “ss,”, “k,”…) and then put them together to determine the word.

Consider two nonsense words, “sklee” and “skloo”. Speak them aloud and notice that for “skloo”, you push your lips forward (so called “rounding” in anticipation of the upcoming “oo”) before you utter the “ss”. Such rounding influences the sound of the “ss,” lowering the frequency spectrum compared to the “ss” sound in “sklee” – a phenomenon known as anticipatory coarticulation. Thus, the “oo” phoneme reveals its presence in the “ss” *earlier* than the “k” and “l” which normally occur *before* the “oo” itself! How do we segment the “oo” phoneme from the others when they are so manifestly intermingled? Or should we even try? Perhaps we are focusing on groupings of the wrong size, and that the most useful unit for recognition is somewhat larger.

Closely related to the problem of segmentation is the problem of recognizing or grouping together the various parts of a composite object. The letter “i” or the symbol “**=**” have two connected components, but we see them as one symbol. We effortlessly read a simple word such as **BEATS**. But consider this: Why didn’t we read instead other words that are perfectly good subsets of the full pattern, such as **BE**, **BEAT**, **EAT**, **AT**, and **eats**? Why don't they enter our minds, unless explicitly brought to our attention? Or when we saw the **B** why didn't we read a **P** or an **I**, which are "there" within the **B**? Conversely, how is it that we can read the two unsegmented words in **POLOPONY –** without placing the *entire* input into a single word category?

This is the problem of *subsets* and *supersets* – formally part of *mereology*, the study of part / whole relationships. It appears as though the best classifiers try to incorporate as much of the input into categorization as "makes sense," but not too much. How can this be done automatically?

1.4.3 Classification

The task of the classifier component of a full system is to use the feature vector provided by the feature extractor to assign the object to a category. The abstraction provided by the feature vector representation of the input data enables the development of a largely domain – independent theory of classification.

The degree of difficulty of the classification problem depends on the variability in the feature values for objects in the same category relative to the difference between feature values for objects in different categories. The variability of feature values for objects in the same category may be due to complexity, and may be due to noise. Noise is any property of the sensed pattern which is not due to the true underlying model but instead to randomness in the world or the sensors. All nontrivial decision and pattern recognition problems involve noise in some form.

What is the best way to design a classifier to cope with this variability? What is the best performance that is possible?

A classifier is used to recommend actions, for example putting a fish in a specified bucket, each action having an associated cost. The post processor uses the output of the classifier to decide on the recommended action.

The simplest measure of classifier performance is the *classification error rate* – the percentage of new patterns that are assigned to the wrong category. Therefore, it is common to seek minimum-error-rate classification. But it is better to recommend actions that will minimize the total expected cost, which is called the *risk*.

The post processor might also be able to exploit *context* – input-dependent information other than from the target pattern itself – to improve the system performance. Context can be highly complex and abstract. For example, the utterance “jeetyet?” may seem nonsensical, unless you hear it spoken by a friend in the context of a cafeteria at lunchtime – “did you eat yet?” How can such a visual and temporal context influence your recognition of speech?

In our fish example, we saw how using multiple features could lead to improved recognition. We may also expect that there would be an improved performance by using multiple classifiers wherein each classifier would be operating on different aspects of the input. For example, we would combine the results of acoustic recognition and lip reading to improve the performance of a speech recognition system.

Self Assessment Questions

6. Pattern Recognition (PR) is a tool for __________________ problem.
7. The ______________________ approach applies biological concepts to machines to recognize patterns.
8. The input to a pattern recognition system is often some kind of______, such as a camera or a microphone array.
9. This is the problem of *subsets* and *supersets* – formally part of __________, the study of part / whole relationships.
10. ____________ is the use of one or more transformations of the input features to produce new salient features.

1.5 The Design Cycle

The design of a pattern recognition system involves the repetition of the following set of activities:

– Data collection
– Choosing Relevant Features
– Choosing appropriate model
– Training the classifiers
– Evaluating the Classifier(s)

This section gives an overview of the design cycle and discusses some of the problems that frequently arise.

The Fig. 1.3 below illustrates the Pattern Recognition design cycle.

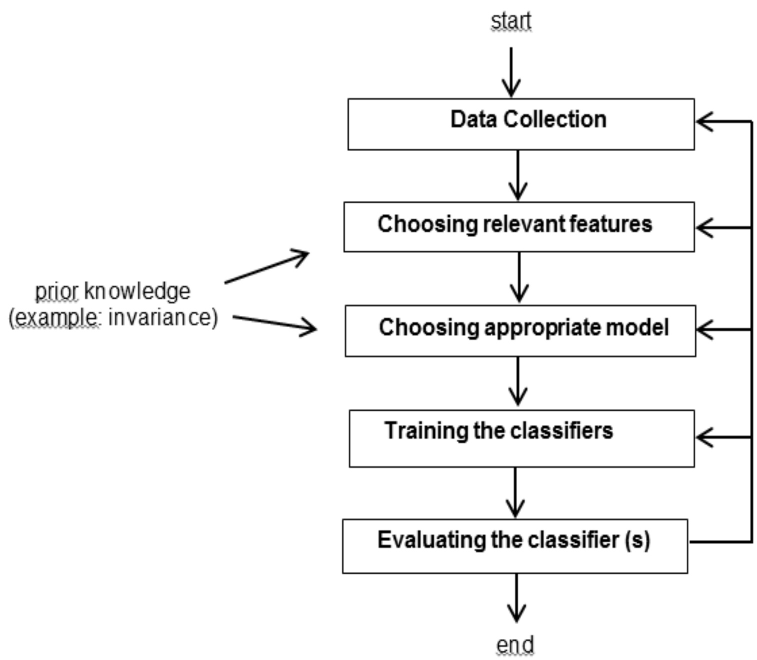

Fig. 1.3: Pattern Recognition Design Cycle

Data Collection: It involves the major cost component in a pattern recognition system. It may be possible to perform a preliminary feasibility study with a small set of "typical" examples, but much more data will usually be needed to assure good performance in the fielded system. We need to collect adequately large and representative set of examples for training and testing the system.

Choosing Relevant Features: The choice of the distinguishing features is a critical design step and depends on the characteristics of the problem domain. Having access to sample data, such as pictures of fish on the conveyor belt will be valuable for choosing a feature set. However, prior knowledge also plays a vital role.

In our hypothetical fish-classification example, prior knowledge about the lightness of different fish categories help in the design of a classifier by suggesting a promising feature. Incorporating prior knowledge can be far more subtle and difficult. In some applications, the knowledge ultimately derives from information about the production of patterns. In others, the knowledge may be about the form of the underlying categories, or specific attributes of the patterns, such as the fact that a face has two eyes, one nose, and so on.

In selecting or designing features, we obviously would like to find features that are simple to extract, invariant or irrelevant transformations, insensitive to noise, and useful for discriminating patterns in different categories.

Choosing Appropriate Model: How do we know when a hypothesized model differs significantly from the true model underlying our patterns and thus a new model is needed? How are we to know to reject a class of models and try another one? Are we as designers reduced to random and tedious trial and error in model selection, never really knowing whether we can expect improved performance? Or might there be principled methods for knowing when to discard one class of models and invoke another?

Training the Classifiers: The process of using data to determine the classifier is referred to as *training* the classifier.

We have already seen many problems that arise in the design of pattern recognition systems. No universal methods have been found for solving all of these problems. However, the repeated experience of the last quarter century has been that the most effective methods for developing classifiers involve learning from example patterns.

Evaluating the Classifiers: When we went from the use of one feature to two in our fish classification problem, it was essentially the result of an evaluation that the error rate we could obtain with one feature was inadequate, and that it was possible to do better. Evaluation is important both to measure the performance of the system and to identify the need for improvements in its components.

The design cycle is repeated from the last step of evaluating the classifier(s) to data collection for either the new data inputs or if there is any necessity for further classification depending on the specific problem or application.

Overfitting: While an overly complex system may allow perfect classification of the training samples, it is unlikely perform well on new patterns. This situation is known as overfitting. One of the most important areas of research in statistical pattern classification is determining how to adjust the complexity of the model – not so simple that it cannot explain the differences between the categories, yet not so complex as to give poor classification on novel patterns. Are there principled methods for finding the best (intermediate) complexity for a classifier?

Computational Complexity: Some pattern recognition problems can be solved using algorithms that are highly impractical. For instance, we might try to hand label all 20 X 20 pixel images with a category label for optical character recognition, and use table lookup to classify incoming patterns. Although we might in theory achieve error – free recognition, the labeling time and storage requirements would be quite prohibitive since it would require a labeling of each of $2^{20X20} \approx 10^{120}$ patterns. Thus the computational resources necessary and the computational complexity of different algorithms are of considerable practical importance.

In general, we may ask how an algorithm scales as a function of the number of feature dimensions, or the number of patterns or the number of categories. What is the tradeoff between computational ease and performance?

Self Assessment Questions

11. The ___________ involves the major cost component in a pattern recognition system.
12. While an overly complex system may allow perfect classification of the training samples, it is unlikely perform well on new patterns; this situation is known as ________.

1.6 Learning and Adaptation

In a broad sense, any method that incorporates information from training samples in the design of a classifier employs learning. Since almost all practical or interesting pattern recognition problems are so hard that we cannot guess the best classification ahead of time, we shall spend the great majority of our time here considering learning.

Creating classifiers then involves posting some general form of model, or form of the classifier, and using training patterns to learn or estimate the unknown parameters of the model.

Learning refers to some form of algorithm for reducing the error on a set of training data.

A range of gradient descent algorithms that later a classifier's parameters in order to reduce an error measure now permeate the field of statistical pattern recognition and these will demand a great deal of our attention. Learning comes in several general forms.

1.6.1 Supervised learning

In automatic pattern recognition, this term refers to the process of designing a pattern classifier by using a primary set of patterns of known class to determine the choice of specific decision making technique for classifying additional similar samples in the future.

In this learning, a teacher provides a category label or cost for each pattern in a training set, and seeks to reduce the sum of the costs for these patterns.

Features of a learning algorithm:

1) Should be powerful enough to learn the solution to a given problem.
2) The algorithm should be stable to parameter variations.
3) We should be able to determine its convergence in a finite amount of time
4) It should scale reasonably with the number of training patterns, the number of input features or the number of categories.
5) The learning algorithm appropriately favors "simple solutions" rather than complicated ones.

1.6.2 Unsupervised learning

In unsupervised learning or clustering there is no explicit teacher, and the system forms clusters or "natural groupings" of the input patterns. "Natural" is always defined explicitly or implicitly in the clustering system itself; and given a particular set of patterns or cost function, different clustering algorithms lead to different clusters. Often the user will set the hypothesized number of different clusters ahead of time.

1.6.3 Reinforcement learning

The most typical way to train a classifier is to present an input, compute its tentative category label, and use the known target category label to improve the classifier.

For example, in Optical Character Recognition (OCR), the input might be an image of character, the actual output of the classifier the category label "R", and the desired output a "B".

In *reinforcement learning* or *learning with a critic*, no desired category signal is given; instead, the only teaching feedback is that the tentative category is right or wrong.

This is analogous to a critic who merely states that something is right or wrong, but does not say specifically how it is wrong. In pattern classification, it is most common that such reinforcement is binary – either the tentative decision is correct or it is not. How can the system learn from such non-specific feedback?

Self Assessment Questions

13. ____________ refers to some form of algorithm for reducing the error on a set of training data.
14. In ______________ or clustering there is no explicit teacher, and the system forms clusters or "natural groupings" of the input patterns.

1.7 Summary

Let us sum up the important concepts discussed in this unit:

- **Machine Perception:** This section discussed the importance of designing and building machines that can recognize patterns.
- **Pattern Recognition Systems:** Pattern Recognition is the act of taking raw data and making an action based on the 'category' of the pattern. This section discussed the working of pattern recognition systems along with the components involved.
- **The Design Cycle:** This section discussed the following phases of the PR design cycle:
  - Data Collection
  - Feature Choice
  - Model Choice
  - Training
  - Evaluation
  - Computational Complexity
- **Learning and Adaptation:** Creating classifiers involves posting some general form of model, or form of the classifier, and using training

patterns to learn or estimate the unknown parameters of the model. Learning refers to some form of algorithm for reducing the error on a set of training data. This section discussed the types of Learning like Supervised, Unsupervised and Reinforcement.

1.8 Terminal Questions

1. Explain the following terms associated with Pattern Recognition:
   - Features
   - Collection
   - Class or Label
   - Pattern
   - Instance or Exemplar
   - Pattern Recognition System
   - Training
2. Explain the process followed by a Pattern Recognition System for processing the input data.
3. Describe the design cycle of a pattern recognition system with the help of a suitable diagram.

1.9 Answers

Self Assessment Questions

1. Features
2. Training
3. Feature Vector
4. segmentation
5. cost function
6. machine intelligence
7. neural
8. transducer
9. mereology
10. Feature extraction
11. Data Collection
12. overfitting
13. Learning
14. unsupervised learning

Terminal Questions

1. Refer Section 1.2

2. Pattern Recognition is the act of taking raw data and making an action based on the 'category' of the pattern. In pattern recognition, classes or categories indicate the groupings into which the samples are pushed according to the measurements or criteria used.

   The following are the two key approaches to pattern recognition:

   1) Classification – we already know the classes. This indicates that the classes have been labeled or determined prior to classification.

   2) Clustering – we don't know the kinds of classes. This indicates that the classes would be determined post classification process. (Refer Section 1.4)

3. The design of a pattern recognition system involves the repetition of the following set of activities:
   - Data collection
   - Choosing Relevant Features
   - Choosing appropriate model
   - Training the classifiers
   - Evaluating the Classifier(s) (Refer Section 1.5)