UNIT 13

PATTERN CLASSIFICATION

Structure **Page No.**

13.1 INTRODUCTION

Pattern recognition is the discipline of building machines to perform perceptual tasks which humans are particularly good at. e.g. recognize faces, voice, identify species of flowers identify diseases in plants by observing the leaves, identify animals, etc. There are practical needs to find more efficient ways of doing things. e.g. read hand-written symbols, diagnose diseases, identify incoming missiles from radar or sonar signals. The machines may perform these tasks faster, more accurately and cheaply.

The goal of pattern recognition is to clarify complicated mechanisms of decision making processes and automatic these functions using computers.

And now, we will list the objectives of this unit. After going through the unit, please read this list again make sure you have achieved the objectives.

Objectives

After studying this unit, you should be able to

- define pattern recognition
- apply different types of classifiers
- describe discriminant Functions (linear and non-linear)

- use Bayesian classification.

We begin the unit by discussing some basic concepts in the following section.

13.2 BASIC CONCEPTS

Although humans can perform some of the perceptual tasks with ease, there is not sufficient understanding to duplicate the performance with a computer. Because the complex nature of the problems, many pattern recognition research has been concerned with more moderate problems of pattern classification — the assignment of a physical object or event to one of several pre-specified categories.

For example, a lumber mill producing assorted hardwoods wants to automate the process of sorting finished lumber according to the species of trees. Optical sensing is used to distinguish birch lumber from ash lumber. A camera takes pictures of the lumber and passes to on to a feature extractor.

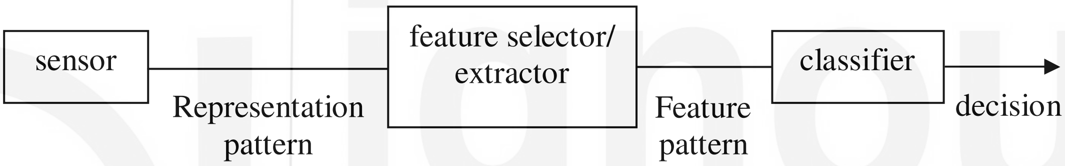


Fig. 1: A pattern classification system

The feature extractor reduces the quantum of data by measuring certain relevant properties that distinguish pictures of birch lumber from pictures of ash lumber. These features are then passed to a classifier that evaluates the evidence presented and makes a final decision about the lumber type.

Suppose that somebody at the lumber mill tells us that birch is often lighter colored than ash. Then brightness becomes an obvious feature. We might attempt to classify the lumber merely by seeing whether or not the average brightness ‘x’ exceeds some critical value.

One characteristic of human pattern recognition is that it involves a teacher. Similarly a machine pattern recognition system needs to be trained. A common mode of learning is to be given a collection of labeled examples, known as training data set. From the training data set, structure information is distilled and used for classifying new inputs.

Try an exercise.

E1) Explain a pattern classification system.

Goal of pattern recognition is to reach an optimal decision rule to categorize the incoming data into their respective categories. A pattern recognition investigation may consist of several stages, enumerated below. Not all stages may be present; some may be merged together so that the distinction between two operations may not be clear, even if both are carried out; also, there may be some application-specific data processing that may not be regarded as one of the stages listed. However, the points below are fairly typical.

1. **Formulation of the problem:** gaining a clear understanding of the aims of the investigation and planning the remaining stages.
2. **Data collection:** making measurements on appropriate variables and recording details of the data collection procedure (ground truth).
3. **Initial examination of the data:** checking the data, calculating summary statistics and producing plots in order to get a feel for the structure.
4. **Feature selection or feature extraction:** selecting variables from the measured set that are appropriate for the task. These new variables may be obtained by a linear or nonlinear transformation of the original set (feature extraction). To some extent, the division of feature extraction and classification is artificial.
5. S**upervised pattern classification:** The system is trained by providing the relevant feature samples with each sample labeled as belonging to a particular class. This help to develop the classifier.
6. **Apply discrimination:** or regression procedures as appropriate. The classifier is designed using a training set of exemplar patterns.
7. **Assessment of results:** This may involve applying the trained classifier to an independent test set of labeled patterns.
8. **Interpretation:** Result interpretation is very important. In case, results are not satisfactory, cycle may be started again from data collection.

We now define the basic terms which are used in the process of pattern recognition.

Feature

Feature is a property (or characteristics) of an object (quantifiable or non quantifiable) which is used to distinguish between (or classify) two different objects. For example, sorting incoming fish on a conveyor according to species using optical sensing. Two category of spices are there: Sea bass, Salmon.

Some properties that could be possibly used to distinguish between the two types of fishes are

• Length
• Lightness (Dark colour or light colour)
• Width

• Number and shape of fins
• Position of the mouth, etc…

This is the set of all suggested features to explore for use in the classifier.

Feature Vector

A Single feature may not be useful always for classification. A set of features used for classification form a feature vector. For example, here the relevant feature vector could be

Fish $x^T = [x1, x2] =$ [Lightness, Width]

Feature Space

The samples of input (when represented by their features) are represented as points in the feature space. If a single feature is used, then the feature space is one- dimensional feature space shown in fig. 2. If number of features is 2, then we get points in 2D space as shown in the Fig. 3. We can also have an n-dimensional feature space.

Fig. 2: Point Representing Samples in 1-D Space

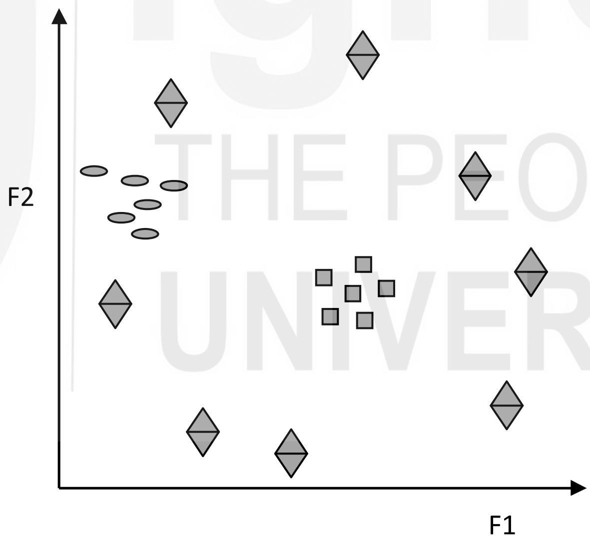
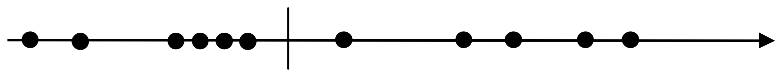


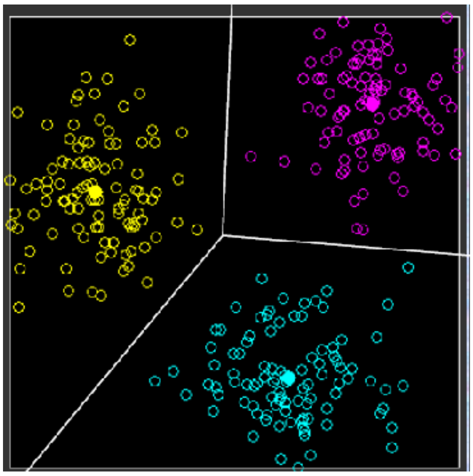
Fig. 3: Sample Points in a 2-Dimensional Feature Space

Decision Region and Decision Boundary

As we know that of pattern recognition is to reach an optimal decision rule to categorize the incoming data into their respective categories (classes). The decision boundary separates points belonging to one class from points of other. The decision boundary partitions the feature space into decision regions. The nature of the decision boundary is decided by the discriminant function which is used for decision. It is a function of the feature vector.



(a) Decision boundary in one-dimensional case with two classes



b) Decision Boundary in 2 (or 3-Dimensional Case with Three Classes

Fig. 4

Hyper Planes and Hyper Surfaces

For two category case, the decision boundary is a line $g(x) = 0$, where $g(x)$ is the discriminant function. When the coordinates of a point are substituted in $g(x)$ and this results in a positive value, we say it belongs to class 1, while a negative value decides the other class. If the number of dimensions is three, then the decision boundary will be a plane or a 3-D surface. The decision regions become semi-infinite volumes. If the number of dimensions increases to more than three, then the decision boundary becomes a hyper-plane or a hyper-surface. The decision regions become semi-infinite hyperspaces.

Learning

The classifier to be designed is built using input samples which is a mixture of all the classes. The classifier learns how to discriminate between samples of different classes. In supervised learning, the classifier is first given a set of training samples and the optimal decision boundary found, and then the classification is done.

Error

The accuracy of classification depends on two things, which are

i) The optimality of decision rule used: The central task is to find an optimal decision rule which categories the training samples correctly, and can generalise to correctly categorising unseen samples as far as possible. This decision theory leads to a minimum error-rate classification.

ii) The accuracy in measurements of feature vectors: This inaccuracy is because of presence of noise. Hence the classifier should deal with noisy and missing features too.

There are various types of classifiers used. We define them as following:

a) **Nonparametric:** Nonparametric techniques do not rely on a set of parameters/weights.

b) **Parametric:** These models are parameterized, with its parameters/weights to be determined through some parameter optimization algorithm, which are then determined by fitting the model to the training data set.

c) **Supervised:** The training samples are given as some input/output pairs. The output is the desired response for the input. The parameters/weights are adjusted so as to minimize the errors between the response of the networks and the desired response.

d) **Unsupervised:** Suppose that we are given data samples without being told which classes they belong to. There are schemes that are aimed to discover significant patterns in the input data without a teacher (labeled data samples).

Try the following exercises.

E2) What is a feature space?

E3) You are given set of data $S =${dolphin, Pomeranian dog, humming bird, frog, rattlesnake, bat}. Developing a suitable classification strategy of classify the given set S according to their nature of existence.

In the following section, we shall discuss discriminant functions.

13.3 DISCRIMINANT FUNCTIONS

Discriminant functions (DFs) are useful for representing classifiers in a simpler way. If we have a set of K classes then we may define a set of K discriminant functions $y_k(x)$, one for each class. Data point 'x' is assigned to class 'c' if

$$y_c(x) > y_k(x) \qquad \text{for all } k \neq c.$$

Formulating a pattern classification problem in terms of discriminant functions is useful since it is possible to estimate the discriminant functions directly from data. Direct estimation of the decision boundaries is sometimes referred to as discriminative modeling. The choice of discriminant function may depend on prior knowledge about the patterns to be classified or may be a particular functional form whose parameters are adjusted by a training procedure. Many different forms of discriminant function have been considered in the literature, varying in complexity from the linear discriminant function (in which g is a linear combination of the x_i) to multi-parameter nonlinear functions such as the multilayer perceptron.
Here, we will discuss some discriminant functions.

Linear Discriminant Functions (LDF)

If no probability distribution or parameters are known, we can to estimate parameters of the discriminant function with Labeled data. The shape of discriminant functions is known such as shown in Fig. 5. If we have samples from 2 classes $x_1, x_2, \ldots, x_n$, we assume that 2 classes can be separated by a linear boundary l (θ) with some unknown parameters θ. Fit the "best" boundary to data by optimizing over parameters θ by minimizing classification error on training data as shown in Fig. 6.

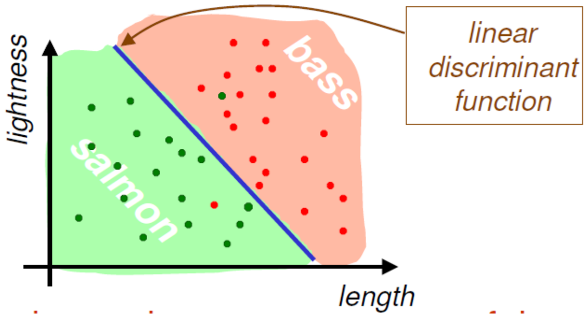


Fig. 5: Linear Discriminant Function

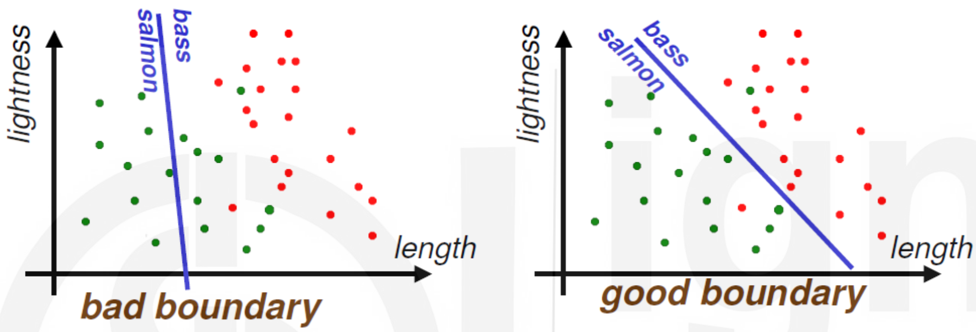


Fig. 6: Linear Discriminant Function for 2 Classes Showing 2 Boundaries

There are many different ways to representa two class pattern classifier. One way is in terms of a discriminant function $g(x)$.

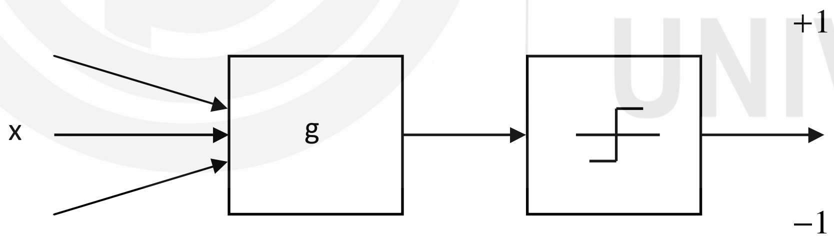


Fig. 7: Block Diagram for Linear Discriminant Function

For a new sample x and a given discriminant function, we can decide on x belongs to Class 1, if $g(x) > 0$, otherwise it belongs to class 2. A discriminant function that is a linear combination of the components of x can be written as

$$g(x) = w^T x + W_0,$$

where w is called the weight vector and w_0 the threshold weight (also referred to as bias). These are the parameters that we want to estimate (learn) based on training data. A classifier based entirely on linear discriminant functions is called a linear classifier or a linear machine.

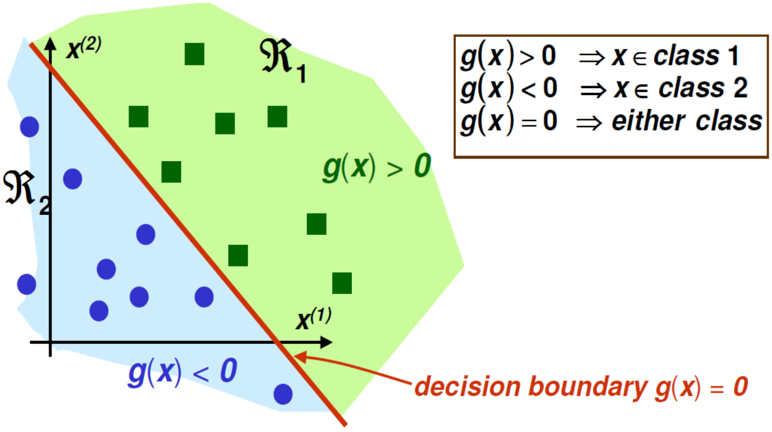
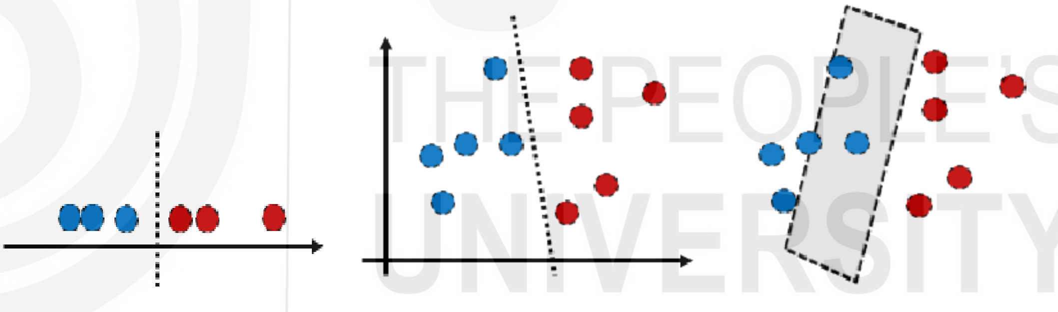


Fig. 8: Linear Discriminant Function for 2 Classes

Decision boundary surface that separates data samples assigned to Class 1 from data samples assigned to Class 2 is given by $g(x) = w^T x + w_0 = 0.$

The equation $g(x) = 0$ defines the decision boundary. This is a hyperplane when $g(x)$ is linear. Set of vectors x are chosen, for some scalars $a_0, \ldots, a_d$ satisfy $a_0 + a_1 x(1) + \cdots + a_d x(d) = 0.$ A hyperplane is a point in 1-D, a line in 2-D, a plane in 3-D as shown in Fig. 9.

Let us consider the family of discriminant functions that are linear combinations of the components of $x = [x_1, \ldots ;; x_p]^T$,



(a) 1-D (b) 2-D (c) 3-D

Fig. 9: Discriminant Function $g(x)$

You may see clearly that in Fig. 9(a) the discriminant function is simply a cut off, and in Fig. 9(b), the discriminant function is a line and in Fig. 9(c), the discriminant function is a plane.

$$g(x) = w^T x + w_0 = \sum_{i=1}^{p} w_i x_i + w_0$$

This is a linear discriminant function, a complete specification of which is achieved by prescribing the weight vector w and threshold weight w_0. Equation of $g(x)$ is the equation of a hyperplane with unit normal in the direction of w and a perpendicular distance $|w_0|/|w|$ from the origin. The value of the discriminant function for a pattern x is a measure of the perpendicular distance from the hyperplane shown in Fig. 10.

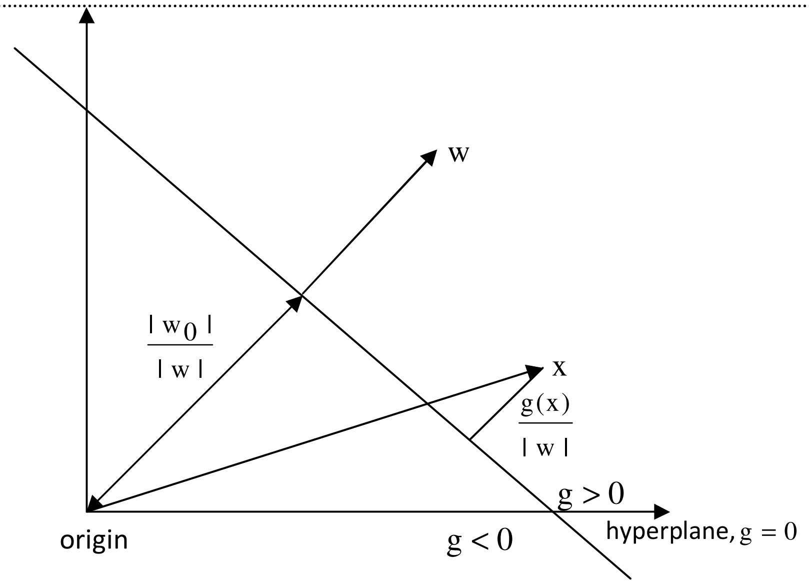


Fig. 10: Geometry of Linear Discriminant Function

Some of the properties of LDA are listed below:

- LDA assumes that the data follows a Gaussian distribution. More specifically, it assumes that all classes share the same covariance matrix.
- LDA finds linear decision boundaries in a $K-1$ dimensional subspace. As such, it is not suited if there are higher-order interactions between the independent variables.
- LDA is well-suited for multi-class problems but should be used with care when the class distribution is imbalanced because the priors are estimated from the observed counts. Thus, observations will rarely be classified to infrequent classes.
- Similarly to Principle Component Analysis (PCA), LDA can be used as a dimensionality reduction technique. Note that the transformation of LDA is inherently different to PCA because LDA is a supervised method that considers the outcomes.

The following are the situation, in which Linear Discriminant Analysis is useful.

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.
- If n is small and the distribution of the predictors X is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.
- Linear discriminant analysis is popular when there are more than two response classes.

Let us understand this through the following example.

Example 1: In order to select the best candidates, an over-subscribed secondary school sets an entrance exam on two subjects of English and Mathematics. The marks of 5 applicants as listed in the Table 1 below and the decision for acceptance is passing an average mark of 75.

(i) Show that the decision rule is equivalent of the method of linear discriminant function.

(ii) Plot the decision hyperplane, indicating the half planes of both Accept and Reject, and location of the 5 applicants.

Table 1

Candidate No.	English	Math	Decision
1	80	85	Accept
2	70	60	Reject
3	50	70	Reject
4	90	70	Accept
5	85	75	Accept

Solution: i) Denote marks of English and Math as x_1 and x_2, respectively. The decision rule is as follows:

If $x_1 + x_2 > 75$, accept, otherwise reject.

This is equivalent to using a linear discriminant function. $g(x) = x_1 + x_2 - 150$ with the following decision rule:

If $g(x) > 0$, accept, otherwise reject.

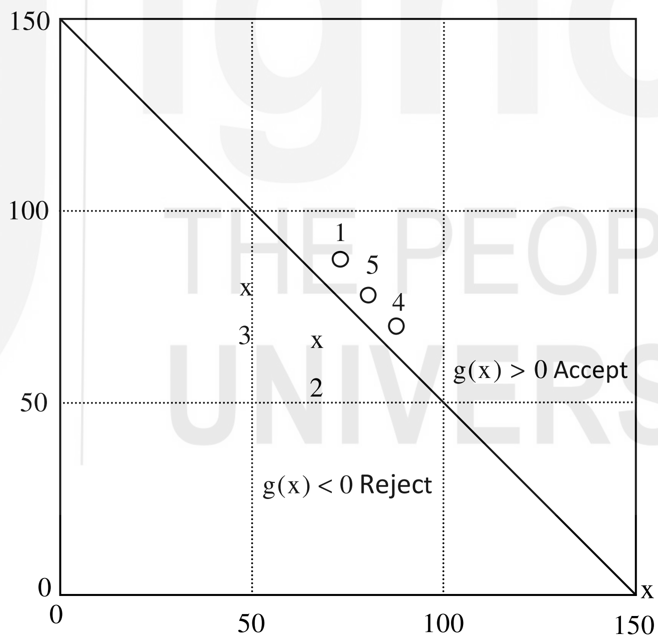


Fig. 11

(ii) To plot $g(x) = 0$, the easiest way is to set $x_1 = 0$, and find the value of x_2 so that $g(x) = 0$.

For this, $0 = 0 + x_2 - 150$, so $x_2 = 150$.
$[0,150]^T$ is on the hyperplane.

Likewise we can also set $x_2 = 0$, find the value of x_1 so that $g(x) = 0$.
i.e. $0 = x_1 + 0 - 150$, so, $x_1 = 150$. $[150, 0]^T$ is on the hyperplane.
Plot a straight line linking $[0,150]^T$ and $[150, 0]^T$ as shown in Fig. 11.

***

Next, we shall discuss another discriminant function.

Piecewise Linear discriminant Functions

Suppose we have 'm' classes, define m linear discriminant functions

$$g_i(x) = w_j^t x + w_{i0} \qquad i = 1, \ldots, m$$

Given x, assign class c_i, if

$$g_i(x) \geq g_j(x) \qquad \forall j \neq i$$

Such classifier is called a linear machine that divides the feature space into c decision regions, with $g_i(x)$ being the largest discriminant if x is in the region R_i.

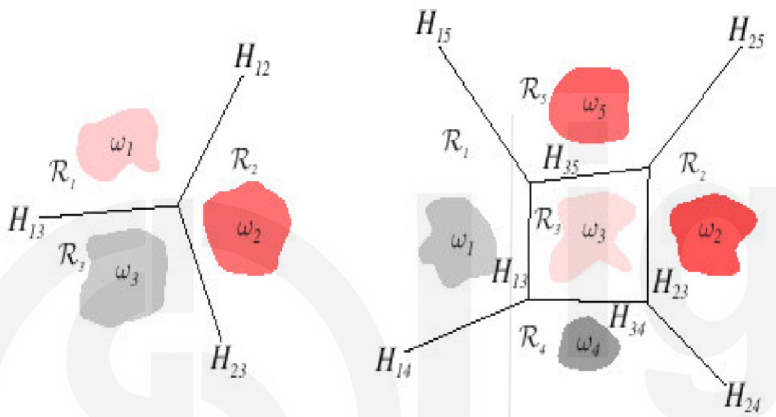


Fig. 12: Linear Discriminant Function for Multiple Classes

For a two contiguous regions R_i and R_j; the boundary that separates them is a portion of hyperplane H_{ij} defined by:

$$g_i(x) = g_j(x) \Leftrightarrow w_i^t x + w_{i0} = w_j^t x + w_{j0}$$
$$\Leftrightarrow (w_i - w_j)^t x + (w_{i0} - w_{j0}) = 0$$

Thus $w_i - w_j$ is normal to H_{ij}. The distance from x to H_{ij} is given by

$$d(x, H_{ij}) = \frac{g_i(x) - g_j(x)}{\|w_i - w_j\|}.$$

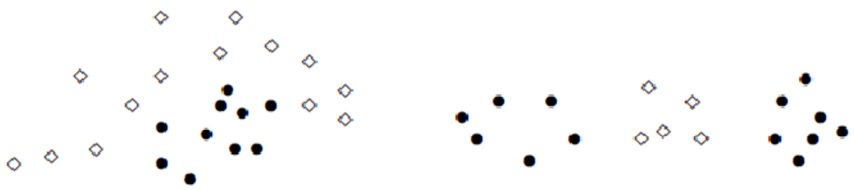


Fig. 13: Groups not Separable by a Linear Discriminant

In a multi-class problem, a pattern x is assigned to the class for which the discriminant function has the maximum value. A linear discriminant function divides the feature space by a hyperplane whose orientation is determined by the weight vector w and distance from the origin by the weight threshold w_0.

Next discriminant function is quadratic discriminant function.

Quadratic Discriminant Function

A quadratic discriminant function is a mapping
$g: X \to R$ with $g(x) = \frac{1}{2} x^T W x + w^T x + w_0$.

In quadratic discriminant function, the model parameter is $\theta = \{W; w, w_0\}$. Depending on W, the geometry of g could be convex, concave, or neither. Fig. 14 shows a quadratic discriminant function separating an inner and an outer cluster of data points.

A quadratic discriminant function is able to classify data using quadratic surfaces. This example shows an ellipsoid surface for separating an inner and outer cluster of data points. QDF is not really that much different from LDF except that you assume that the covariance matrix can be different for each class and so, we will estimate the covariance matrix separately for each class $k, k, = 1, 2, \ldots, K$.

$$\delta_k(x) = -\frac{1}{2}\log\left|\sum_k\right| - \frac{1}{2}(x-\mu_k)^T \sum_k^{-1}(x-\mu_k) + \log \pi_k$$

Classification rule:

$$\hat{G}(x) = \arg\max_k \delta_k(x)$$

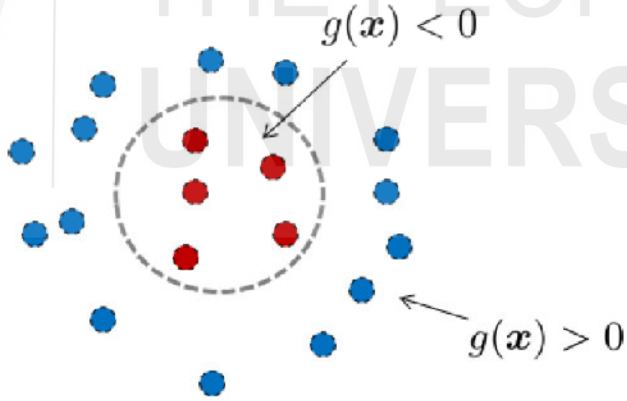


Fig. 14: Quadratic Discriminant Function

The classification rule is similar as well. You just find the class k which maximizes the quadratic discriminant function. The decision boundaries are quadratic equations in x. QDF allows more flexibility for the covariance matrix, tends to fit the data better than LDF, but then it has more parameters to estimate. The number of parameters increases significantly with QDF as a separate covariance matrix is required for every class. If you have many classes and not so many sample points, this can be a problem.

After quadratic discriminant function, let us now discuss the non-linear discriminant function.

Now, try the following exercises.

E4) Explain Linear Discremination Function.

E5) What are the properties of LDA?

In the following section, we shall discuss Bayesian classification.

13.4 BAYESIAN CLASSIFICATION

Goal of most classification procedures is to estimate the probabilities that a pattern to be classified belongs to various possible classes, based on the values of some feature or set of features.

Here, we are discussing Bayesian decision making or Bayes Classifier. This method refers to choosing the most likely class, given the value of the feature/s. Bayes theorem calculates the probability of class membership. In most cases, we decide which is the most likely class. We need a mathematical decision making algorithm, to obtain classification.

Let us recall Bayes theorem used in probability. Accordingly to it, $P(w_i \mid \vec{X}) = \dfrac{P(\vec{X} \mid w_i) P(w_i)}{P(\vec{X})}$, where $P(w_i)$ is the prior probability for class $w_i, P(X)$ is the probability (Unconditional) for feature vector $X, P(w_i \mid X)$ is measured-conditioned or posteriori probability and $P(X \mid w_i)$ is the probability (Class-conditional) of feature vector X in class w_i.

$P(X)$ is the probability distribution for feature X in the entire population. It is also called unconditional density function. $P(w_i)$ is the prior probability that a random sample is a member of the class C_i. $P(X \mid w_i)$ is the class conditional probability (or likelihood) of obtaining feature value X given that the sample is from class w_i. It is equal to the number of times (occurrences) of X, if it belongs to class w_i. **The goal** is to measure $P(w_i \mid X)$, posteriori probability, from the above three values. This is the probability of any vector X being assigned to class w_i.

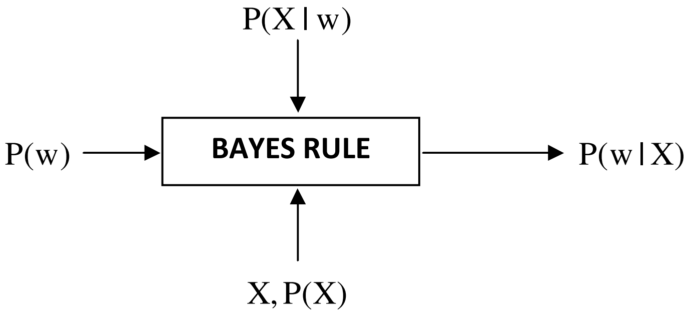


Fig. 15: Bayes Theorem

Here is another model, called the naive Bayes probabilistic model. The probability model for a classifier is a conditional model

$$P(X \mid W_1, \ldots, W_n)$$

over a dependent class variable X with a small number of outcomes or classes, conditional on several feature variables W_1 through W_n. The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes' theorem, we write

$$P(X \mid W_1, \ldots, W_n) = \frac{p(X)\,p(W_1, \ldots, W_n \mid X)}{p(W_1, \ldots, W_n)}$$

In simple words, $\text{posterior} = \dfrac{\text{prior x likelihood}}{\text{evidence}}$

In practice, we are only interested in the numerator of that fraction, since the denominator does not depend on X and the values of the features W_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model $p(X \mid W_1, \ldots, W_n)$, which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$\begin{aligned} p(X \mid W_1, \ldots, W_n) &= p(X)\,p(W_1, \ldots, W_n \mid X_n) \\ &= p(X)\,p(W_1 \mid X)\,p(W_2, \ldots, W_n \mid X, W_1) \\ &= p(X)\,p(W_1 \mid X)\,p(W_2 \mid X, W_1)\,p(W_3, \ldots, W_n \mid X, W_1, W_2) \\ &= p(X)\,p(W_1 \mid X)\,p(W_2 \mid X, W_1), \ldots, \\ &\quad p(W_n \mid X, W_1, W_2, \ldots, W_{n-1}) \end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature W_i is conditionally independent of every other feature W_i for $i \neq j$. This means that

$$p(W_i \mid X, W_j) = p(W_i \mid X)$$

for $i \neq j$ and the joint expression can be expressed as

$$\begin{aligned} p(X \mid W_i, \ldots, W_n) &= p(X)\,p(W_1 \mid X)\,p(W_2 \mid X)\ldots \\ &= p(C)\prod_{i=1}^{n} p(w_i \mid X). \end{aligned}$$

This means that under the above independence assumptions, the conditional distribution over the class variable can be expressed like this:

$$p(X \mid W_1, \ldots, W_n) = \frac{1}{z}\,p(C)\prod_{i=1}^{n} p(w_i \mid X),$$

where Z (the evidence) is a scaling factor dependent only on $W_i, \ldots W_n$, i.e., a constant, if the values of the feature variables are known.

Let us now discuss how Parameter estimation is done.

All model parameters (i.e., class priors and feature probability distributions) can be approximated with relative frequencies from the training set. These are maximum likelihood estimates of the probabilities. A class' prior may be calculated by assuming equiprobable classes (i.e., priors = 1 / (number of classes)), or by calculating an estimate for the class probability from the training set (i.e., (prior for a given class) = (number of samples in the class) / (total number of samples)).

To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set. If one is dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

For example, suppose the training data contains a continuous attribute, x. We first segment the data by the class and then compute the mean and variance of x in each class. Let μ_c be the mean of the values in x associated with class c, and let $\sigma_c{}^2$ be the variance of the values x in associated with class c. Then, the probability of some value given a class, $p(x = v \mid c)$, can be computed by plugging into the equation for a Normal distribution parameterized by μ_c and $\sigma_c{}^2$. That is,

$$p(x = v \mid c) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(v-\mu_c^2)^2}{2\sigma_c^2}}$$

Another common technique for handling continuous values is to use binning to discretize the values. In general, the distribution method is a better choice if there is a small amount of training data, or if the precise distribution of the data is known. The discretization method tends to do better if there is a large amount of training data because it will learn to fit the distribution of the data. Since naive Bayes is typically used when a large amount of data is available (as more computationally expensive models can generally achieve better accuracy), the discretization method is generally preferred over the distribution method.

If a given class and feature value never occurs together in the training set then the frequency-based probability estimate will be zero. This is problematic since it will wipe out all information in the other probabilities when they are multiplied. It is therefore often desirable to incorporate a small-sample correction in all probability estimates such that no probability is ever set to be exactly zero.

Now, we shall Construct a classifier from the probability model.
The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the *maximum a posteriori* or *MAP* decision rule. The corresponding classifier is the function classify defined as follows:

$$\text{classify}(W_1, \ldots, W_n) = \arg\max p(C = c) \prod_{i=1}^{n} p(W_i = w_i \mid C = c)$$

Despite the fact that the far-reaching independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution.

This in turn helps to alleviate problems stemming from the curse of dimensionality, such as the need for data sets that scale exponentially with the number of features. Like all probabilistic classifiers under the MAP decision rule, it arrives at the correct classification as long as the correct class is more probable than any other class; hence class probabilities do not have to be estimated very well. In other words, the overall classifier is robust enough to ignore serious deficiencies in its underlying naive probability model.

Properties of Bayes Classifiers

1. **Incrementality:** with each training example, the prior and the likelihood can be updated dynamically. It is flexible and robust to errors.
2. **Combines prior knowledge and observed data:** prior probability of a hypothesis is multiplied with probability of the hypothesis given the training data.
3. **Probabilistic hypotheses:** outputs is not only a classification, but a probability distribution over all classes.
4. **Meta-classification:** the outputs of several classifiers can be combined, e.g., by multiplying the probabilities that all classifiers predict for a given class.

Now, let us see the following examples.

Example 2: (Two class problem): Let us define variables, Cold (C) and not-cold (C'). Feature is fever (f). Prior probability of a person having a cold, $P(C) = 0.01$. Prob. of having a fever, given that a person has a cold is, $P(f \mid C) = 0.4$. Overall prob. of fever $P(f) = 0.02$. Find the Prob. that a person has a cold, given that she (or he) has a fever.

Solution: Using Bayes theorem, the Prob. that a person has a cold, given that she (or he) has a fever is:

$$P(C \mid f) = \frac{p(f \mid c)\, P(c)}{P(f)} = \frac{0.4 * 0.001}{0.02} = 0.2$$

let us take an example with values to verify:

Total Population =1000.

Thus, people hav ing cold = 10.

People having both fever and cold = 4.

Thus, people having only cold = 10 – 4 = 6.

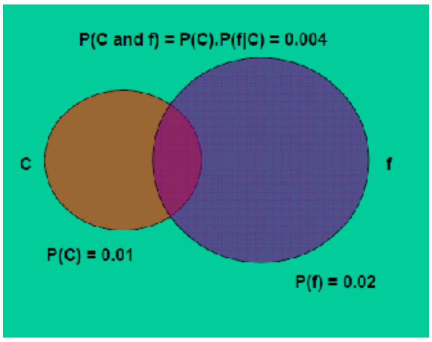


Fig. 15: A Venn Diagram

People having fever (with and without cold) = 0.02 * 1000 = 20.

People having fever without cold = 20 – 4 = 16

So, probability (percentage) of people having cold along with fever, out of all those having fever, is $= 4/20 = 0.2(20\%)$.

Probability of a joint event - a sample comes from class C and has the feature value X:

$$P(C \text{ and } X) = P(C).P(X \mid C)$$
$$= 0.01 * 0.4$$

Or, $P(C \text{ and } X) = P(X).P(C \mid X)$
$$= 0.02 * 0.2$$

Also verify, for a K class problem:

$$P(X) = P(w_1)P(X \mid w_1) + P(w_2)P(X \mid w_2) + \cdots + P(w_k)P(X \mid w_k)$$
$$P(w_i \mid \vec{X}) = \frac{P(\vec{X} \mid w_i)P(w_i)}{P(w_1)P(X \mid w_1) + P(w_2)P(X \mid w_2) + \cdots + P(w_k)P(X \mid w_k)}$$

We get
$$P(f) = P(C)P(f \mid C) + P(C')P(f \mid C')$$
$$= 0.01 * 0.4 + 0.99 * 0.01616 = 0.02$$

***

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Try following exercises.

E6) Explain Bayes classifier.

E7) Explain properties of Bayes classifier.

Let us now summarize, what we have discussed in this unit.

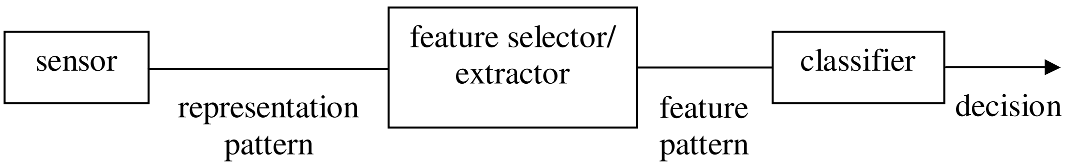
13.5 SUMMARY

We have discussed the following points.

1) Concept of Classification.
2) Various terminologies in pattern recognition problem.
3) Linear discriminant function, piecewise discriminant function, quadratic discrimint function and non-linear discriminant function.
4) Bayes classifier combines prior knowledge with observed data: assigns a posterior probability to a class based on its prior probability and its likelihood given the training data. It computes the maximum a posterior (MAP) hypothesis or the maximum likelihood (ML) hypothesis.
5) Naive Bayes classifier assumes conditional independence between attributes and assigns the MAP class to new instances.
6) Likelihoods can be estimated based on frequencies. Sparse data poses a huge problem. A possible solution is to use m-estimate.

13.6 SOLUTION/ANSWERS

E1)

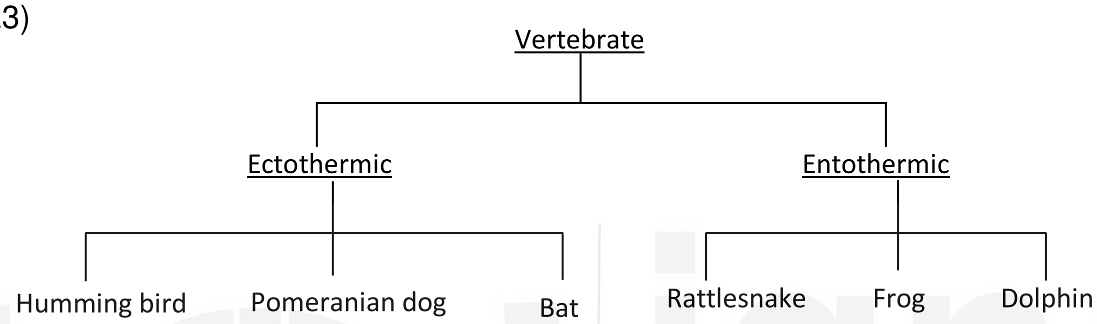


Optical sensing is used to distinguish two patterns. A camera takes pictures of the object and passes to on to a feature extractor. The feature extractor reduces the data by measuring certain “properties” that distinguish pictures of one object to the other. These features are then passed to a classifier that evaluates the evidence presented and makes a final decision about the object type.

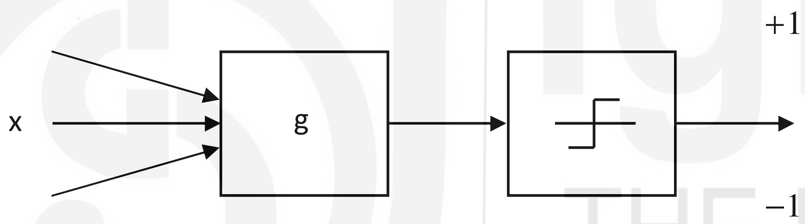
One characteristic of human pattern recognition is that it involves a teacher. Similarly a machine pattern recognition system needs to be trained. A common mode of learning is to be given a collection of labeled examples, known as training data set. From the training data set, structure information is distilled and used for classifying new inputs.

E2) The samples of input (when represented by their features) are represented as points in the feature space. If a single feature is used, then work on a one- dimensional feature space. If number of features is 2, then we get points in 2D space. We can also have an n-dimensional feature space.

E3)



E4)



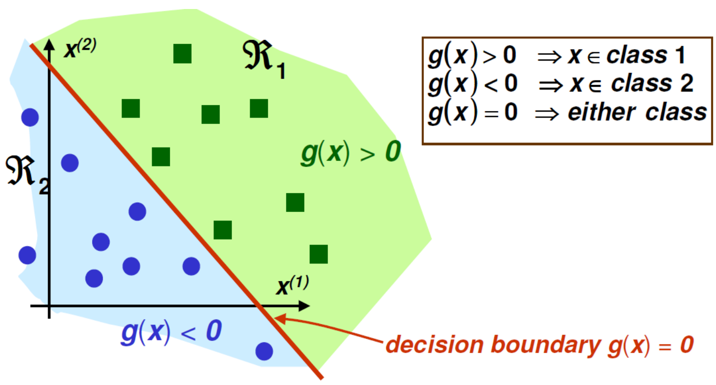
For a new sample x and a given discriminant function, we can decide on x belongs to

Class1 if $g(x) > 0$, otherwise it's Class 2.

A discriminant function that is a linear combination of the components of x can be written as

$$g(x) = w^T x + w_0$$

where w is called the weight vector and w_0 the threshold weight. These are the parameters that we want to estimate based on training data. A classifier based entirely on linear discriminant functions is called a linear classifier or a linear machine.



E5) LDF assumes that the data are Gaussian. More specifically, it assumes that all classes share the same covariance matrix.

- LDF finds linear decision boundaries in a $K-1$ dimensional subspace. As such, it is not suited if there are higher-order interactions between the independent variables.
- LDF is well-suited for multi-class problems but should be used with care when the class distribution is imbalanced because the priors are estimated from the observed counts. Thus, observations will rarely be classified to infrequent classes.
- Similarly to PCA, LDA can be used as a dimensionality reduction technique. Note that the transformation of LDA is inherently different to PCA because LDA is a supervised method that considers the outcomes.

E6) The naive Bayes classifier combines bayes model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the *maximum a posteriori* or *MAP* decision rule. The corresponding classifier is the function classify defined as follows:

$$\text{classify}(W_1, \ldots, W_n) = \arg\max p(C = c) \prod_{i=1}^{n} p(W_i = w_i \mid C = c)$$

Despite the fact that the far-reaching independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution.

E7)

- **Incrementality:** with each training example, the prior and the likelihood can be updated dynamically. It is flexible and robust to errors.
- **Combines prior knowledge and observed data:** prior probability of a hypothesis is multiplied with probability of the hypothesis given the training data.
- **Probabilistic hypotheses:** outputs is not only a classification, but a probability distribution over all classes.
- **Meta-classification:** the outputs of several classifiers can be combined, e.g., by multiplying the probabilities that all classifiers predict for a given class.