Pattern Recognition

P.S.Sastry

sastry@ee.iisc.ernet.in

Dept. Electrical Engineering

Indian Institute of Science, Bangalore

Reference Books

- R.O.Duda, P.E.Hart and D.G.Stork, ‘Pattern Classification’, Johy Wiley, 2002
- C.M.Bishop, ‘Pattern Recognition and Machine Learning’, Springer, 2006.
- C. M. Bishop, ‘Neural Networks for Pattern Recognition’, Oxford University Press, Indian Edition, 2003.
- R.O.Duda and P.E. Hart, ‘Pattern Classification and Scene Analysis’, Wiley, 1973

Pattern Recognition

A basic attribute of people – categorisation of sensory input

Examples of Pattern Recognition tasks

- ‘Reading’ facial expressions
- Recognising Speech
- Reading a Document
- Identifying a person by fingerprints
- Diagnosis from medical images
- Wine tasting

Machine Recognition of Patterns

- Feature extractor makes some measurements on the input pattern.
- X is called *Feature Vector*. Often, $X \in \Re^n$.
- Classifier maps each feature vector to a class label.
- Features to be used are problem-specific.

Some Examples of PR Tasks

- Character Recognition
  - Pattern – Image.
  - Class – identity of character
  - Features: Binary image, projections (e.g., row and column sums), Moments etc.

Some Examples of PR Tasks

- Speech Recognition
  - Pattern – 1-D signal (or its sampled version)
  - Class – identity of speech units
  - Features – LPC model of chunks of speech, spectral info, cepstrum etc.
  - Pattern can become a *sequence* of feature vectors.

Examples contd...

- Fingerprint based identity verification
  - Pattern – image plus a identity claim
  - Class – Yes / No
  - Features: Position of Minutiae, Orientation field of ridge lines etc.

Examples contd...

- Video-based Surveillance
  - Pattern – video sequence
  - Class – e.g., level of alertness
  - Features – Motion trajectories, Parameters of a prefixed model etc.

Examples contd...

- Credit Screening
  - Pattern – Details of an applicant (for, e.g., credit card)
  - Class – Yes / No
  - Features: income, job history, level of credit, credit history etc.

Examples contd...

- Imposter detection (of, e.g., credit card)
  - Pattern – A sequence of transactions
  - Class – Yes / No
  - Features: Amount of money, locations of transactions, times between transactions etc.

Examples contd...

- Document Classification
  - Pattern – A document and a query
  - Class – Relevant or not (in general, rank)
  - Features – word occurrence counts, word context etc.
- Spam filtering, diagnostics of machinery etc.

Design of Pattern Recognition Systems

- Features depend on the problem. Measure ‘relevant’ quantities.

Design of Pattern Recognition Systems

- Features depend on the problem. Measure ‘relevant’ quantities.
- Some techniques available to extract ‘more relevant’ quantities from the initial measurements. (e.g., PCA)

Design of Pattern Recognition Systems

- Features depend on the problem. Measure ‘relevant’ quantities.
- Some techniques available to extract ‘more relevant’ quantities from the initial measurements. (e.g., PCA)
- After feature extraction each pattern is a vector
- Classifier is a function to map such vectors into class labels.

Design of Pattern Recognition Systems

- Features depend on the problem. Measure ‘relevant’ quantities.
- Some techniques available to extract ‘more relevant’ quantities from the initial measurements. (e.g., PCA)
- After feature extraction each pattern is a vector
- Classifier is a function to map such vectors into class labels.
- Many general techniques of classifier design are available.
- Need to test and validate the final system.

Some notation

- Feature Space, $\mathcal{X}$ – Set of all possible feature vectors.

Some notation

- Feature Space, $\mathcal{X}$ – Set of all possible feature vectors.
- Classifier: a decision rule or a function, $h : \mathcal{X} \rightarrow \{1, \cdots, M\}$.
- Often, $\mathcal{X} = \Re^n$. Convenient to take $M = 2$. Then we take the labels as $\{0, 1\}$ or $\{-1, 1\}$.

Some notation

- Feature Space, $\mathcal{X}$ – Set of all possible feature vectors.
- Classifier: a decision rule or a function, $h : \mathcal{X} \to \{1, \cdots, M\}$.
- Often, $\mathcal{X} = \Re^n$. Convenient to take $M = 2$. Then we take the labels as $\{0, 1\}$ or $\{-1, 1\}$.
- Then, any binary valued function on $\mathcal{X}$ is a classifier.
  - What h to choose? We want **correct** or **optimal** classifier.
  - How to design classifiers?
  - How to judge performance?
  - How to provide performance guarantees?

- We first consider the 2-class problem.

- We first consider the 2-class problem.
- Can handle the $M > 2$ case if we know how to handle 2-class problem.

- We first consider the 2-class problem.
- Can handle the $M > 2$ case if we know how to handle 2-class problem.
- Simplest alternative: design M 2-class classifiers. ‘One Vs Rest’

- We first consider the 2-class problem.
- Can handle the $M > 2$ case if we know how to handle 2-class problem.
- Simplest alternative: design M 2-class classifiers. ‘One Vs Rest’
- There are other possibilities: e.g., Tree structured classifiers.

- We first consider the 2-class problem.
- Can handle the $M > 2$ case if we know how to handle 2-class problem.
- Simplest alternative: design M 2-class classifiers. ‘One Vs Rest’
- There are other possibilities: e.g., Tree structured classifiers.
- The 2-class problem is the basic problem.

- We first consider the 2-class problem.
- Can handle the $M > 2$ case if we know how to handle 2-class problem.
- Simplest alternative: design M 2-class classifiers. ‘One Vs Rest’
- There are other possibilities: e.g., Tree structured classifiers.
- The 2-class problem is the basic problem.
- We will also look at M-class classifiesrs.

A simple PR problem

- : Problem: ‘Spot the Right Candidate’
- : Features:
  - x_1: Marks based on academic record
  - x_2: Marks in the interview

A simple PR problem

- : Problem: ‘Spot the Right Candidate’
- : Features:
  - x_1: Marks based on academic record
  - x_2: Marks in the interview
- A Classifier: $ax_1 + bx_2 > c \Rightarrow$ ‘Good’

A simple PR problem

- : Problem: ‘Spot the Right Candidate’
- : Features:
  - x_1: Marks based on academic record
  - x_2: Marks in the interview
- A Classifier: $ax_1 + bx_2 > c \Rightarrow$ ‘Good’
  Another Classifier: $x_1 x_2 > c \Rightarrow$ ’Good’
  (or $(x_1 + a)(x_2 + b) > c$).

A simple PR problem

- : Problem: ‘Spot the Right Candidate’
- : Features:
  - x_1: Marks based on academic record
  - x_2: Marks in the interview
- A Classifier: $ax_1 + bx_2 > c \Rightarrow$ ‘Good’
  Another Classifier: $x_1 x_2 > c \Rightarrow$ ’Good’
  (or $(x_1 + a)(x_2 + b) > c$).
- Design of classifier:
  We have to choose a specific form for the classifier.
  What values to use for parameters such as a, b, c?

Designing Classifiers

- Need to decide how feature vector values determine the class.
  (How different marks reflect goodness of candidate)

Designing Classifiers

- Need to decide how feature vector values determine the class.
  (How different marks reflect goodness of candidate)
- In most applications, not possible to design classifier from ’physics of the problem’.

Designing Classifiers

- Need to decide how feature vector values determine the class.
  (How different marks reflect goodness of candidate)
- In most applications, not possible to design classifier from ’physics of the problem’.
- The difficulties are
  - Lot of variability in patterns of a single class
  - Variability in feature vector values
  - Feature vectors of patterns from different classes can be arbitrarily close.
  - Noise in measurements

Designing Classifiers contd...

- Often the only information available for the design is – A training set of example patterns.
- Training set: $\{(X_i, y_i), \quad i = 1, \ldots, \ell\}$.
  Here X_i is an example feature vector of class y_i.

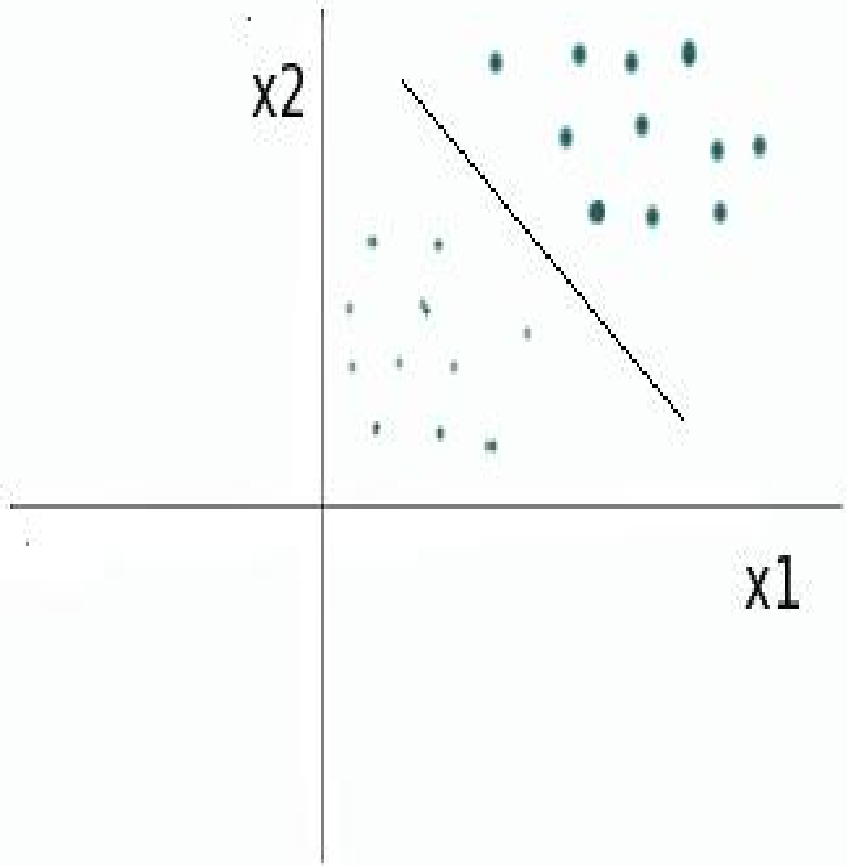
Designing Classifiers contd...

- Often the only information available for the design is – A training set of example patterns.
- Training set: $\{(X_i, y_i), \quad i = 1, \ldots, \ell\}$.
  Here X_i is an example feature vector of class y_i.
- Generation of training set – Take representative patterns of known category (data collection) and obtain the feature vectors. (choice of feature measurements).
- Now *learn* an *appropriate* function h as classifier. (Model choice)
- Test and validate the classifier on more data.

A simple PR problem

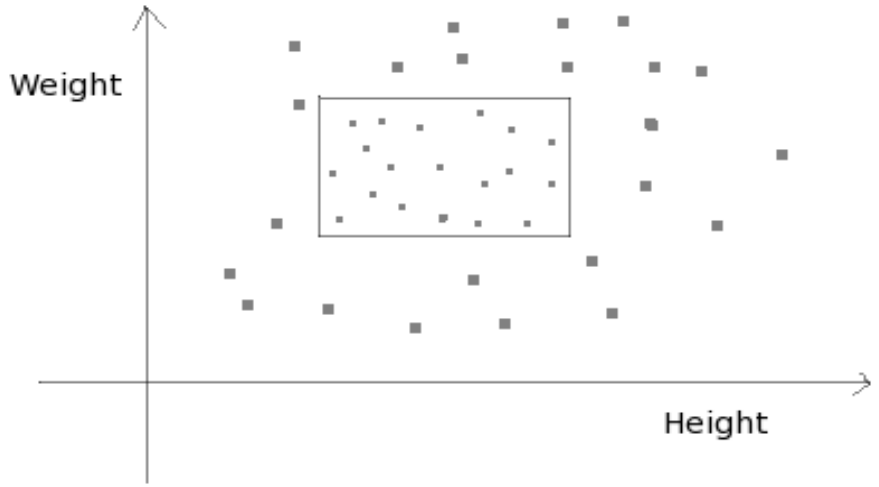
- : Problem: ‘Spot the Right Candidate’
- : Features:
  - x_1: Marks based on academic record
  - x_2: Marks in the interview
- A Classifier: $ax_1 + bx_2 > c \Rightarrow$ ‘Good’
  We have chosen a specific form for the classifier.
- Design of classifier: What values to use for a, b, c?
- **Information available: ‘experience’ – history of past candidates**

Training Set



Another example problem

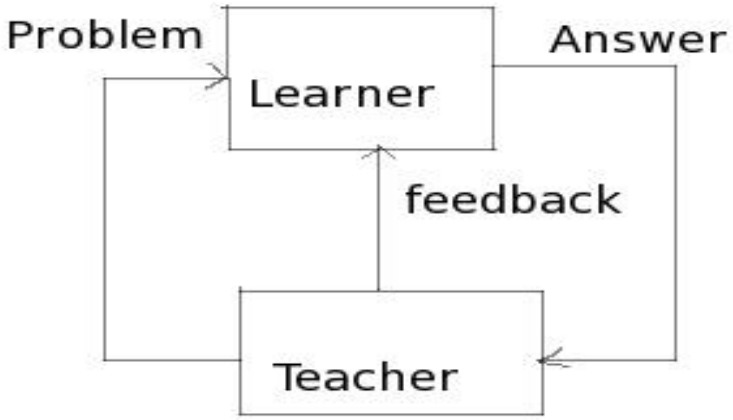
- Problem: recognize persons of ‘medium build’
- Features: Height and Weight



The classifier is ‘nonlinear’ here

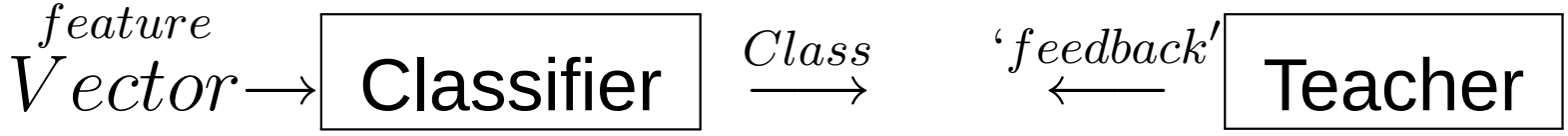
Learning from Examples

- Designing a classifier is a typical problem of learning from examples. (Also called learning with a teacher).



- Nature of feedback from teacher determines difficulty of the learning problem.

In the context of Pattern Recognition



- *Supervised learning* – Teacher gives the true class label for each feature vector
- *Reinforcement Learning* – noisy assessment of performance, e.g., correct/incorrect
- *Unsupervised learning* – no teacher input (Clustering problems)

- When the class labels of training patterns as given by ‘teacher’ are noisy, we consider it as supervised learning with label noise or classification noise.
- Many classifier design algorithms do supervised learning.

Function Learning

- Closely related problem. Output is continuous-valued rather than discrete as in classifiers.
- Here training set examples could be $\{(X_i, y_i), \;\; i = 1, \ldots, \ell\}$, $X_i \in \mathcal{X}$, $y_i \in \Re$.

Function Learning

- Closely related problem. Output is continuous-valued rather than discrete as in classifiers.
- Here training set examples could be $\{(X_i, y_i),\ \ i = 1, \ldots, \ell\}$, $X_i \in \mathcal{X}, y_i \in \Re$.
- The prediction variable, y, is continuous; rather than taking finitely many values. (There can be noise in examples).
- Similar learning techniques needed to infer the underlying functional relationship between X and y. (Regression function of y on X).

Examples of Function Learning

- Time series prediction: Given a series $x_1, x_2, \cdots$, find a function to predict x_n.

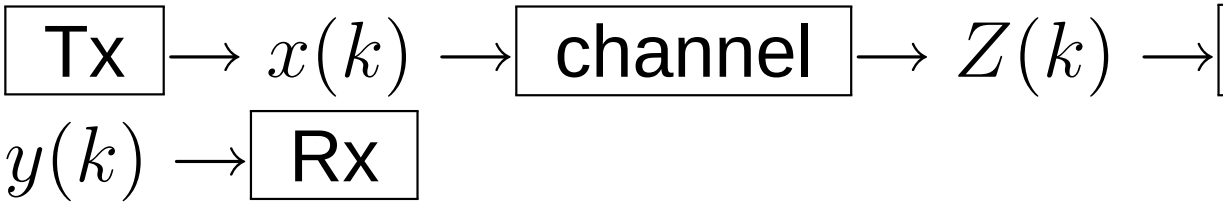
Examples of Function Learning

- Time series prediction: Given a series $x_1, x_2, \cdots$, find a function to predict x_n.
- Based on past values: Find a 'best' function $\hat{x}_n = h(x_{n-1}, x_{n-2}, \cdots, x_{n-p})$
  - Predict stock prices, exchange rates etc.
  - Linear prediction model used in speech analysis

Examples of Function Learning

- Time series prediction: Given a series $x_1, x_2, \cdots$, find a function to predict x_n.
- Based on past values: Find a 'best' function $\hat{x}_n = h(x_{n-1}, x_{n-2}, \cdots, x_{n-p})$
  - Predict stock prices, exchange rates etc.
  - Linear prediction model used in speech analysis
- More general predictors can use other variables also.
  - Predict rainfall based on measurements and (possibly) previous years' data.
  - In general, System Identification. (An application: smart sensors)

Examples contd... : Equaliser



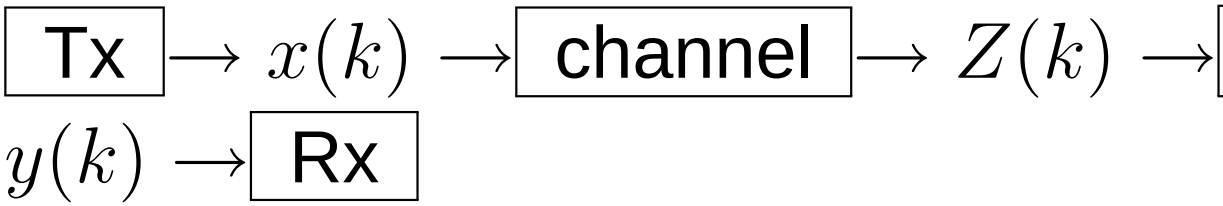
We want $y(k) = x(k)$. Design (or adapt) the filter to achieve this

We can choose a filter as

$$y(k) = \sum_{i=1}^{T} a_i Z(k-i)$$

Find 'best' a_i – a function learning problem.

Examples contd... : Equaliser



We want $y(k) = x(k)$. Design (or adapt) the filter to achieve this

We can choose a filter as

$$y(k) = \sum_{i=1}^{T} a_i Z(k-i)$$

Find 'best' a_i – a function learning problem.

Training set: $\{(x(k), Z(k)), k = 1, 2, \cdots, N\}$

How do we know $x(k)$ at the receiver end?

Prior agreements (Protocols)

Learning from examples

- Learning from examples is basic to both classification and regression.
- Training set: $\{(X_1, y_1), \cdots, (X_\ell, y_\ell)\}$.
- We essentially want to fit a 'best' function: $y = f(X)$.

Learning from examples

- Learning from examples is basic to both classification and regression.
- Training set: $\{(X_1, y_1), \cdots, (X_\ell, y_\ell)\}$.
- We essentially want to fit a 'best' function: $y = f(X)$.
- Suppose $X \in \Re$. Then this is a familiar 'curve-fitting' problem.

Learning from examples

- Learning from examples is basic to both classification and regression.
- Training set: $\{(X_1, y_1), \cdots, (X_\ell, y_\ell)\}$.
- We essentially want to fit a 'best' function: $y = f(X)$.
- Suppose $X \in \Re$. Then this is a familiar 'curve-fitting' problem.
- Model choice (choice of form for f) is very important here.
- If we choose f to be a polynomial, higher the degree 'better' would be the performance on training data.
- But that is not the real performance – A polynomial of degree ℓ would give zero error!!

Learning from examples – Generalization

- To obtain a classifier (or a regression function) we use the training set.
- We ‘know’ the class label of patterns (or the values for prediction variable) in the training set.

Learning from examples – Generalization

- To obtain a classifier (or a regression function) we use the training set.
- We ‘know’ the class label of patterns (or the values for prediction variable) in the training set.
- Errors on the training set do not necessarily tell how good is the classifier.

Learning from examples – Generalization

- To obtain a classifier (or a regression function) we use the training set.
- We ‘know’ the class label of patterns (or the values for prediction variable) in the training set.
- Errors on the training set do not necessarily tell how good is the classifier.
- Any classifier that amounts to only storing the training set is useless.
- Interested in the ‘generalization abilities’ – how does our classifier perform on unseen or new patterns.

Design of Classifiers

- The classifier should perform well inspite of inherent variability of patterns and noise in feature extraction and/or in class labels as given in training set.

Design of Classifiers

- The classifier should perform well inspite of inherent variability of patterns and noise in feature extraction and/or in class labels as given in training set.
- Statistical Pattern Recognition – An approach where the variabilities are captured through probabilistic models.

Design of Classifiers

- The classifier should perform well inspite of inherent variability of patterns and noise in feature extraction and/or in class labels as given in training set.
- Statistical Pattern Recognition – An approach where the variabilities are captured through probabilistic models.
- There are other approaches, e.g., syntactic pattern recognition, fuzzy-set based methods etc.

Design of Classifiers

- The classifier should perform well inspite of inherent variability of patterns and noise in feature extraction and/or in class labels as given in training set.
- Statistical Pattern Recognition – An approach where the variabilities are captured through probabilistic models.
- There are other approaches, e.g., syntactic pattern recognition, fuzzy-set based methods etc.
- In this course we consider classification and regression (function learning) problems in the statistical framework.

Statistical Pattern Recognition

- $\mathcal{X}$ is the feature space. (We take $\mathcal{X} = \Re^n$). We Consider a 2-class problem for simplicity of notation.
- A given feature vector can come from different classes with different probabilities.

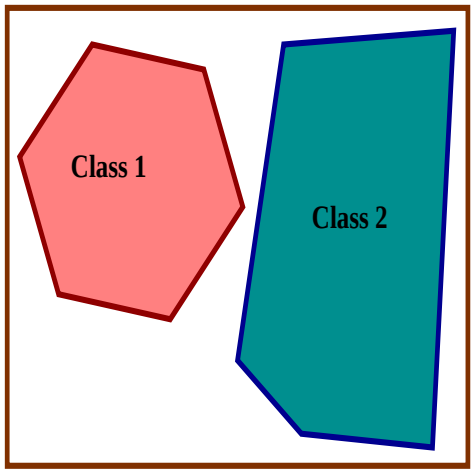
Statistical Pattern Recognition

- $\mathcal{X}$ is the feature space. (We take $\mathcal{X} = \Re^n$). We Consider a 2-class problem for simplicity of notation.
- A given feature vector can come from different classes with different probabilities.
- Let: f_i be the probability density function of the feature vectors from class-i, $i = 0, 1$.
- f_i are called *class conditional densities*.

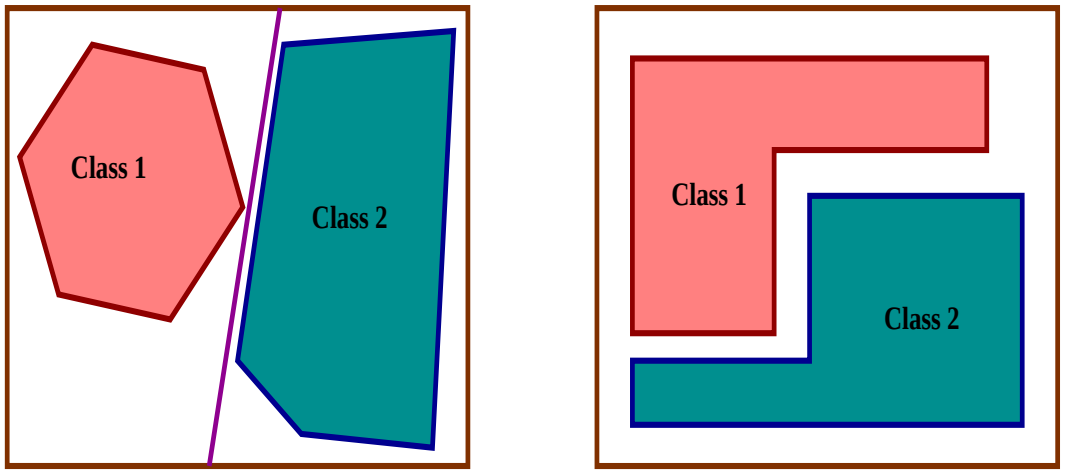
Statistical Pattern Recognition

- $\mathcal{X}$ is the feature space. (We take $\mathcal{X} = \Re^n$). We Consider a 2-class problem for simplicity of notation.
- A given feature vector can come from different classes with different probabilities.
- Let: f_i be the probability density function of the feature vectors from class-i, $i = 0, 1$.
- f_i are called *class conditional densities*.
- Let $\mathbf{X} = (X_1, \cdots, X_n) \in \Re^n$ represent the feature vector.
- Then f_i is the (conditional) joint density of the random variables $X_1, \cdots, X_n$ given that $\mathbf{X}$ is from class-i.

- Class conditional densities model the variability in the feature values.
- For example, the two classes can be uniformly distributed in the two regions as shown. (The two classes are *separable* here).

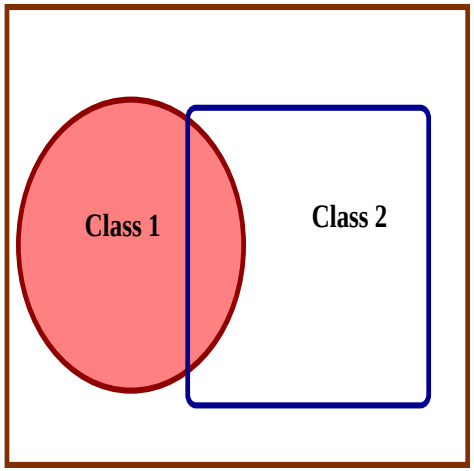


- When class regions are separable, an important special case is linear separability.



- The classes in the left panel above are linearly separable (can be separated by a line) while those in the right panel are not linearly separable (though separable).

- In general, the two class conditional densities can overlap. (The same value of feature vector can be from different classes with different probabilities)



- The statistical viewpoint gives us one way of looking for ‘optimal’ classifier.
- We can say we want a classifier that has least probability of misclassifying a random pattern (drawn from the underlying distributions).

- Let

$$q_i(X) = \text{Prob}[\text{class} = i | X], \; i = 0, 1.$$

q_i is called posterior probability (function) for class-i.

- Let

$$q_i(X) = \text{Prob}[\text{class} = i | X], \; i = 0, 1.$$

  q_i is called posterior probability (function) for class-i.

- Consider the classifier

$$\begin{aligned} h(X) &= 0 \text{ if } q_0(X) > q_1(X) \\ &= 1 \text{ otherwise} \end{aligned}$$

- Let
$$q_i(X) = \text{Prob}[\text{class} = i | X], \; i = 0, 1.$$
q_i is called posterior probability (function) for class-i.
- Consider the classifier
$$\begin{aligned} h(X) &= 0 \text{ if } q_0(X) > q_1(X) \\ &= 1 \text{ otherwise} \end{aligned}$$
- $q_0(X) > q_1(X)$ would imply that the feature vector X is more likely to come from class-0 rather than class-1.

- Let
$$q_i(X) = \text{Prob}[\text{class} = i | X], \; i = 0, 1.$$
q_i is called posterior probability (function) for class-i.
- Consider the classifier
$$\begin{aligned} h(X) &= 0 \text{ if } q_0(X) > q_1(X) \\ &= 1 \text{ otherwise} \end{aligned}$$
- $q_0(X) > q_1(X)$ would imply that the feature vector X is more likely to come from class-0 rather than class-1.
- Hence, intuitively, such a classifier should minimize probability of error in classification.

Statistical Pattern Recognition

- $\mathcal{X}(=\Re^n)$ is the feature space. $\mathcal{Y}=\{0,1\}$ is the set of class labels

Statistical Pattern Recognition

- $\mathcal{X}(=\Re^n)$ is the feature space. $\mathcal{Y}=\{0,1\}$ is the set of class labels
- $\mathcal{H}=\{h \mid h:\mathcal{X}\rightarrow\mathcal{Y}\}$ is the set of classifiers of interest.

Statistical Pattern Recognition

- $\mathcal{X}(=\Re^n)$ is the feature space. $\mathcal{Y}=\{0,1\}$ is the set of class labels
- $\mathcal{H}=\{h \mid h:\mathcal{X}\rightarrow\mathcal{Y}\}$ is the set of classifiers of interest.
- For a feature vector X, let $y(X)$ denote the class label of X. In general, $y(X)$ would be random.

Statistical Pattern Recognition

- $\mathcal{X}(=\Re^n)$ is the feature space. $\mathcal{Y}=\{0,1\}$ is the set of class labels
- $\mathcal{H}=\{h \mid h:\mathcal{X}\rightarrow\mathcal{Y}\}$ is the set of classifiers of interest.
- For a feature vector X, let $y(X)$ denote the class label of X. In general, $y(X)$ would be random.
- Now we want to assign a figure of merit to each possible classifier in $\mathcal{H}$.

- For example, we can rate different classifiers by

$$F(h) = \text{Prob}[h(X) \neq y(X)]$$

- $F(h)$ is the probability that h misclassifies a random X.

- For example, we can rate different classifiers by

$$F(h) = \text{Prob}[h(X) \neq y(X)]$$

- $F(h)$ is the probability that h misclassifies a random X.
- Optimal classifier would be one with lowest value of F.

- For example, we can rate different classifiers by

$$F(h) = \text{Prob}[h(X) \neq y(X)]$$

- $F(h)$ is the probability that h misclassifies a random X.
- Optimal classifier would be one with lowest value of F.
- Given a h we can calculate $F(h)$ *only if* we know the probability distributions of classes.

- For example, we can rate different classifiers by

$$F(h) = \text{Prob}[h(X) \neq y(X)]$$

- $F(h)$ is the probability that h misclassifies a random X.
- Optimal classifier would be one with lowest value of F.
- Given a h we can calculate $F(h)$ *only if* we know the probability distributions of classes.
- Minimizing F is not a straight-forward optimization problem.

- For example, we can rate different classifiers by

$$F(h) = \text{Prob}[h(X) \neq y(X)]$$

- $F(h)$ is the probability that h misclassifies a random X.
- Optimal classifier would be one with lowest value of F.
- Given a h we can calculate $F(h)$ *only if* we know the probability distributions of classes.
- Minimizing F is not a straight-forward optimization problem.
- Here we are treating all errors as same. We will

Statistical PR contd.

- Recall that $f_i(X)$ denotes the probability density function of feature vectors of class-i. (class conditional densities).

Statistical PR contd.

- Recall that $f_i(X)$ denotes the probability density function of feature vectors of class-i. (class conditional densities).
- Let $p_i = \text{Prob}[y(X) = i]$. Called *prior probabilities*.

Statistical PR contd.

- Recall that $f_i(X)$ denotes the probability density function of feature vectors of class-i. (class conditional densities).
- Let $p_i = \text{Prob}[y(X) = i]$. Called *prior probabilities*.
- Recall, *posterior probabilities*, $q_i(X) = \text{Prob}[y(X) = i \mid X]$.

Statistical PR contd.

- Recall that $f_i(X)$ denotes the probability density function of feature vectors of class-i. (class conditional densities).
- Let $p_i = \text{Prob}[y(X) = i]$. Called *prior probabilities*.
- Recall, *posterior probabilities*, $q_i(X) = \text{Prob}[y(X) = i \mid X]$. Now, by Bayes theorem

$$q_i(X) = f_i(X)p_i \ / \ Z$$

where $Z = f_0(X)p_0 + f_1(X)p_1$ is the normalising constant

Bayes Classifier

- Consider the classifier given by

$$h(X) = 0 \text{ if } \frac{q_0(X)}{q_1(X)} > 1$$
$$= 1 \text{ otherwise}$$

Bayes Classifier

- Consider the classifier given by

$$\begin{aligned} h(X) &= 0 \text{ if } \frac{q_0(X)}{q_1(X)} > 1 \\ &= 1 \text{ otherwise} \end{aligned}$$

- This is called the Bayes classifier. Given our statistical framework, this is the optimal classifier.

Bayes Classifier

- Consider the classifier given by

$$\begin{aligned} h(X) &= 0 \text{ if } \frac{q_0(X)}{q_1(X)} > 1 \\ &= 1 \text{ otherwise} \end{aligned}$$

- This is called the Bayes classifier. Given our statistical framework, this is the optimal classifier.
- $q_0(X) > q_1(X)$ is same as $p_0 f_0(X) > p_1 f_1(X)$.

Bayes Classifier

- Consider the classifier given by

$$\begin{aligned} h(X) &= 0 \text{ if } \frac{q_0(X)}{q_1(X)} > 1 \\ &= 1 \text{ otherwise} \end{aligned}$$

- This is called the Bayes classifier. Given our statistical framework, this is the optimal classifier.
- $q_0(X) > q_1(X)$ is same as $p_0 f_0(X) > p_1 f_1(X)$.
- We will prove optimality of Bayes classifier later.

story so far

- We consider PR as a two step process – Feature measurement/extraction and Classification
- A classifier is to map feature vectors to Class labels.
- The main difficulty in designing classifiers is due to large variability in feature vector values.
- The main information we have for the design is a training set of examples.
- Function learning is a closely related problem.
- In both cases we need to learn from (training) examples.

story so far

- The statistical view is: model the variation in feature vectors from a given class through probability models.
- One objective can be: Find a classifier that has least probability of error.
- For example, Bayes Classifier is the optimal one if we know class conditional densities.

Organization of the course

- Overview of classifier learning strategies
- Nearest Neighbour classification rule
- Bayes Classifier for minimizing risk.
- Some variations on this theme (e.g. Neymann-Pearson Classifier)
- Techniques for estimating class conditional densities to implement Bayes classifier.

Organization of the course

- Linear discriminant functions
- Learning linear discriminant functions (Perceptron, LMS, logistic regression)
- Linear least-squares regression
- Simple overview of statistical learning theory
- Empirical risk minimization and VC theory

Organization of the course

- Learning nonlinear classifiers
- Neural Networks (Backpropagation, RBF networks)
- Support Vector Machines
- Kernel-based methods
- Feature extraction and dimensionality reduction (PCA)
- Boosting and ensemble classifiers (AdaBoost)

Thank You!