

# Introduction to Pattern Recognition

Presented By –


Anup Singh

4<sup>th</sup> year, DMS Student

IISER Kolkata

# Outline

---

- What is pattern recognition?
  - Feature Extraction techniques – LDA and PCA
  - Bayes Decision Rule
  - Discriminant functions
  - Results of Classification using Bayes Rule
- 
- A solid green horizontal bar spanning the width of the slide, located at the bottom.

# Human Perception

---

- Humans have developed highly sophisticated skills for sensing their environment and taking actions according to what they observe, e.g.,
  - Recognizing a face.
  - Understanding spoken words.
  - Reading handwriting.
  - Distinguishing fresh food from its smell.
- We would like to give the similar capability to the machines.



# Pattern Recognition (PR)

---

- Pattern Recognition is the study of how machines can:
  - observe the environment,
  - learn to distinguish patterns of interest,
  - make sound and reasonable decisions about the categories of the patterns.

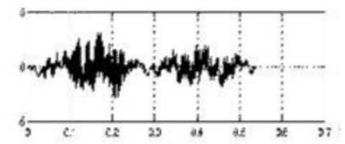


# Pattern Recognition (PR)

- **What is a Pattern?**

- is an abstraction, represented by a set of measurements describing a “physical” object

- Many types of patterns exist:

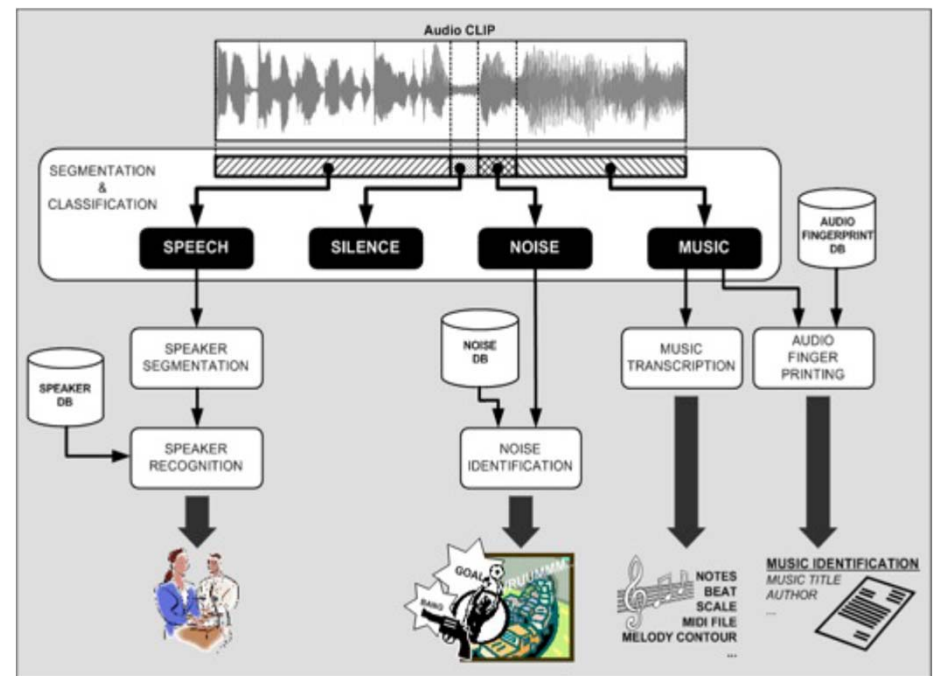


John Smith

# Pattern Recognition (PR)

- **What is a Pattern Class (or category)?**

- is a set of patterns sharing common attributes
- a collection of “similar”, not necessarily identical, objects
- During recognition, given objects are assigned to a prescribed class



# Pattern Recognition (PR)

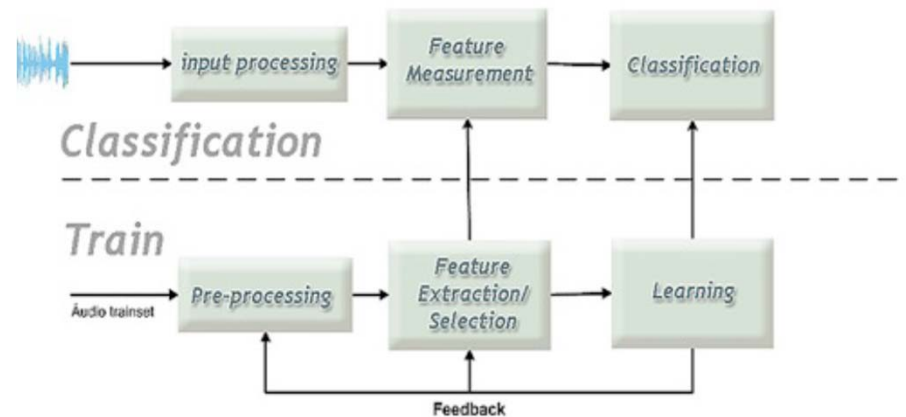
---

- Two phase Process

1. Training/Learning

- System must be exposed to several examples of each class
- Creates a “model” for each class

2. Detecting/Classifying



# Pattern Recognition (PR)

---

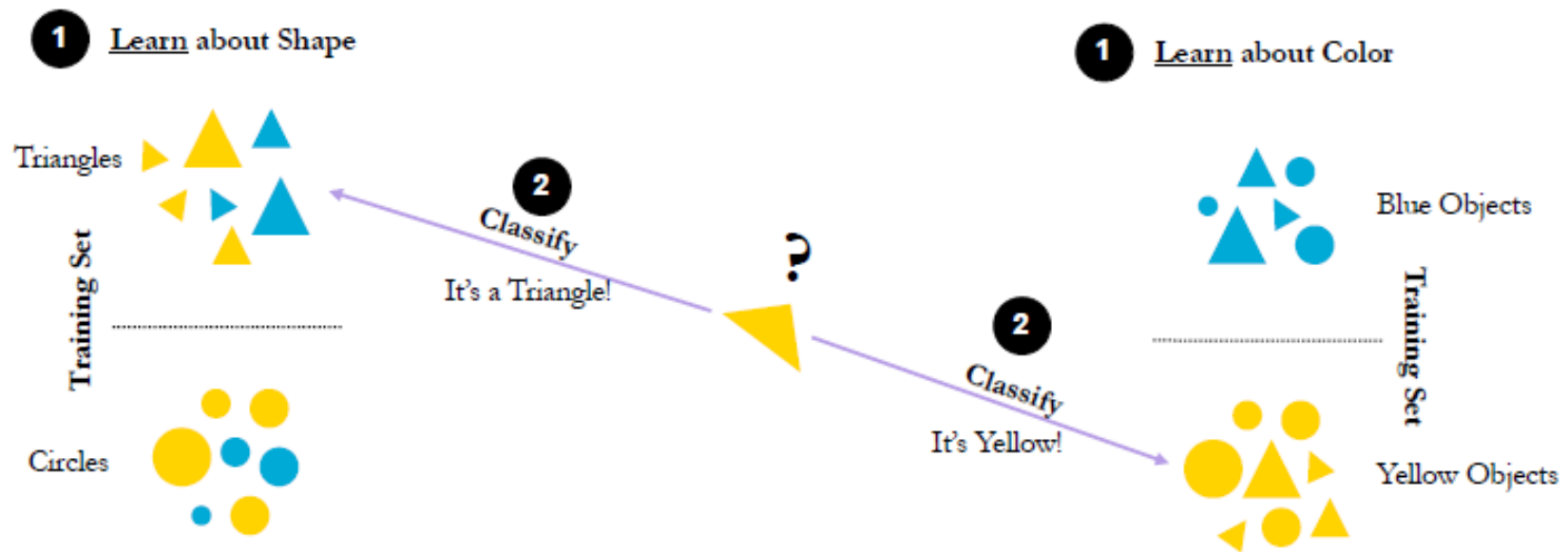
- How can a machine learn the rule from data?
  - **Supervised learning:** a teacher provides a category label for each pattern in the training set.
    - ❑ Classification
  - **Unsupervised learning:** the system forms clusters or natural groupings of the input patterns (based on some similarity criteria).
    - ❑ *Clustering*



# Pattern Recognition (PR)

- **Supervised Training/Learning**

- a “teacher” provides labeled training sets, used to train a classifier

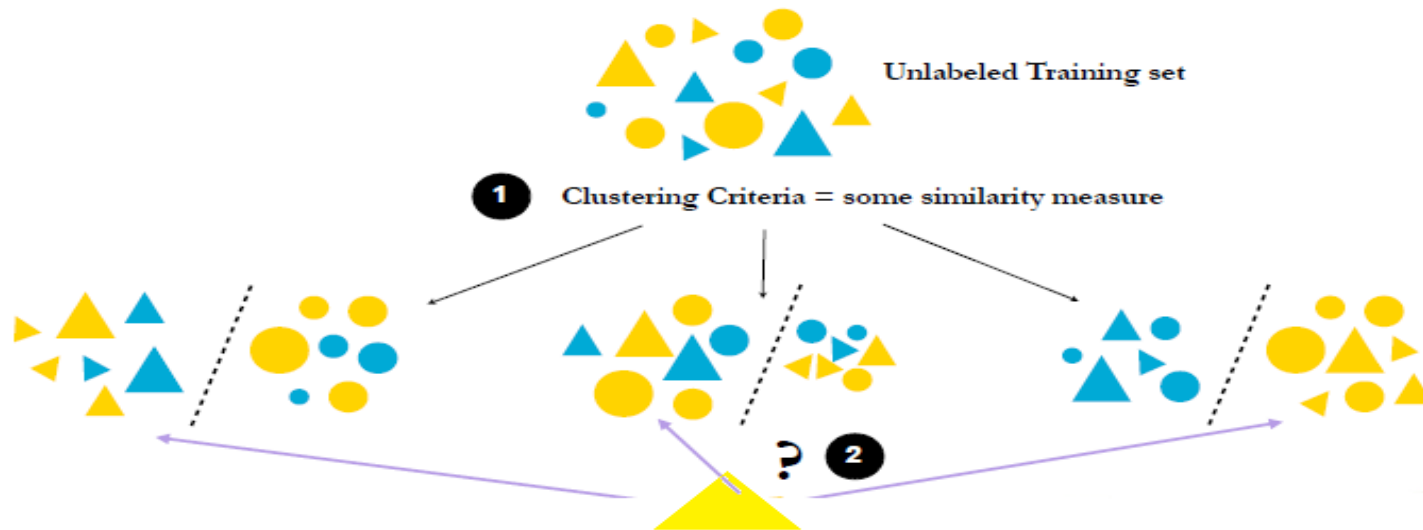


# Pattern Recognition (PR)

---

- **Unsupervised Training/Learning**

- No labeled training sets are provided
- System applies a specified clustering/grouping criteria to unlabeled dataset
- Clusters/groups together “most similar” objects (according to given criteria)



# Pattern Recognition Process

- **Data acquisition and sensing:**

- Measurements of physical variables.
- Important issues: bandwidth, resolution , etc.

- **Pre-processing:**

- Removal of noise in data.
- Isolation of patterns of interest from the background.

- **Feature extraction:**

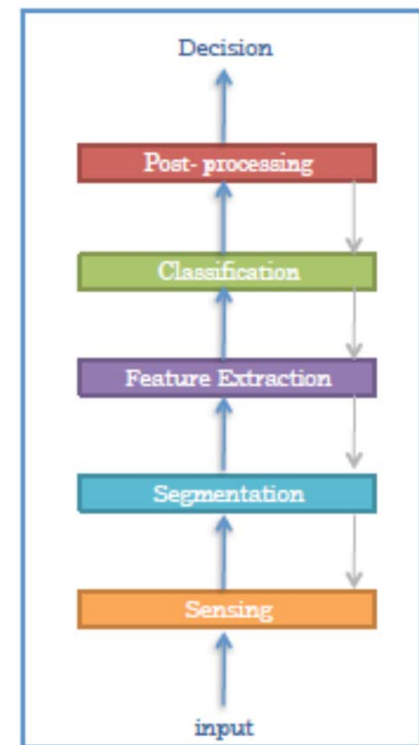
- Finding a new representation in terms of features.

- **Classification**

- Using features and learned models to assign a pattern to a category.

- **Post-processing**

- Evaluation of confidence in decisions.



# Features

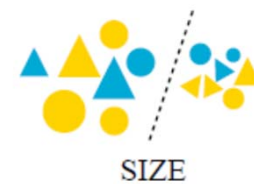
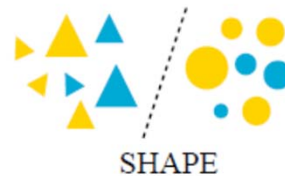
---

- Features are properties of an object
- Paves the way to a good discrimination of different classes of objects!
- Allows to group the objects into different classes

- Take a group of graphical objects:

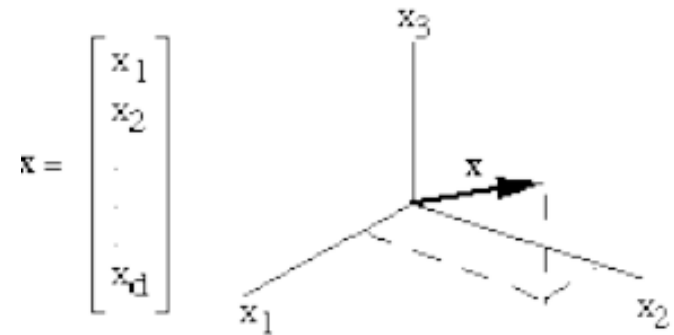
➤ Possible features:

- Shape
- Color
- Size etc.



# Feature Vectors

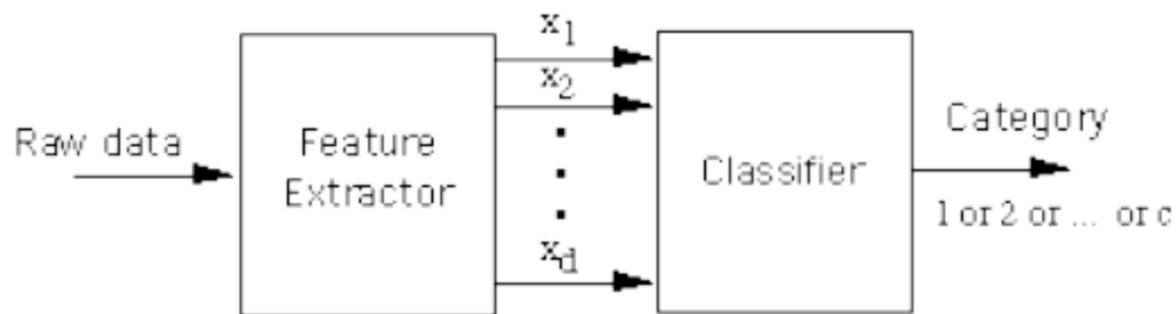
- Usually a single object can be represented using several features, e.g.
  - $x_1$  = shape (e.g. no. of sides)
  - $x_2$  = size (e.g. some numeric value)
  - $x_3$  = color (e.g. rgb values)
  - ....
  - $x_d$  = some other (numeric) feature.
- $\mathbf{x}$  becomes a feature vector
  - $\mathbf{x}$  is a point in a d-dimensional **feature space**.



# The Classical Model for PR

---

- A **Feature Extractor** extracts features from raw data (e.g. audio, image, weather data, etc)
- A **Classifier** receives  $X$  and assigns it to one of  $c$  categories, Class 1, Class 2, ..., Class  $c$  (i.e. labels the raw data).



# The Classical Model for PR

---

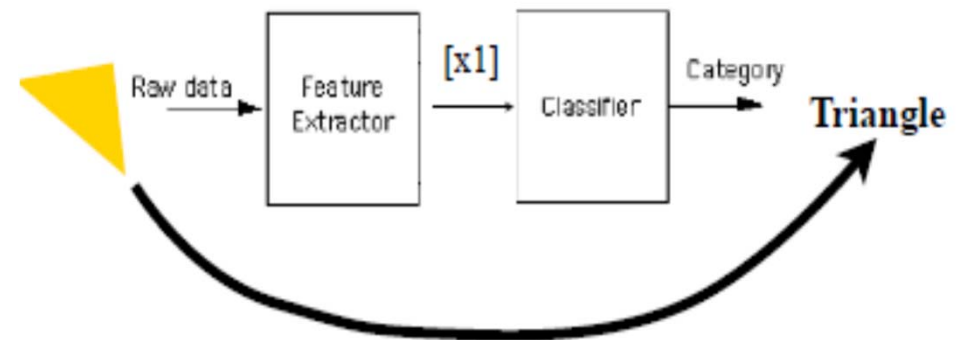
- **Example: classify graphic objects according to their shape**

- **Feature extracted:**

- no. of sides (x1)

- **Classifier:**

- 0 sides => circle
    - 3 sides => triangle
    - (4 sides => rectangle)



# Feature Extraction

---

- **Principal Component Analysis(PCA)**

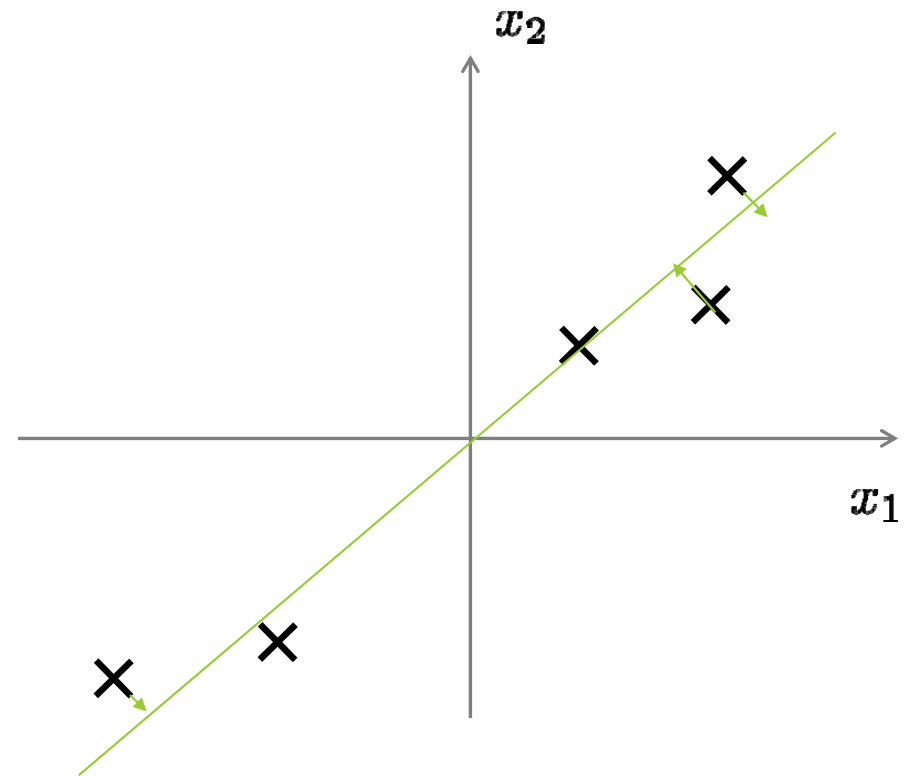
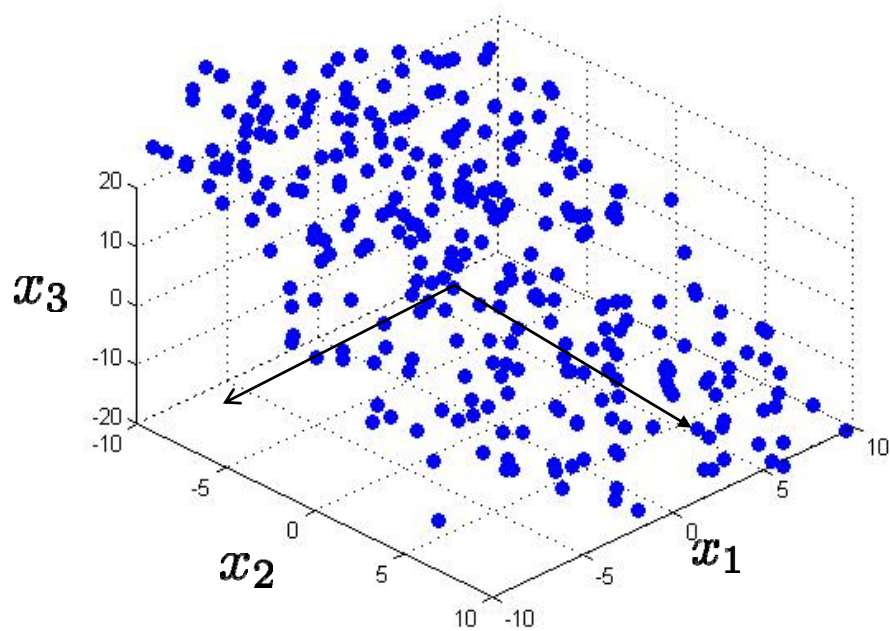
- transforms a number of possibly correlated variables(features) into a smaller number of variables(features) called principal components.
- PCA reduces the dimensionality of the data by retaining as much as variation possible in our original data set.

- **Linear Discriminant Analysis(LDA)**

- Tries to find a linear combination of features that characterizes or separates two or more classes of objects or events.



# PCA is not linear regression



# Data preprocessing:

Training set:  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}$

Preprocessing (feature scaling/mean normalization):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}$$

Replace each  $x_j^{(i)}$  with  $x_j - \mu_j$

If different features on different scales (e.g.,  $x_1$  = size of house,  $x_2$  = number of bedrooms), scale features to have comparable range of values.

# Principal Component Analysis (PCA) algorithm

- Reduce data from n-dimensions to k-dimensions
- Compute “covariance matrix”:

$$\Sigma = \frac{1}{m} \sum_{i=1}^n (x^{(i)})(x^{(i)})^T$$

- Compute “eigenvectors” of matrix  $\Sigma$  :

$$[U, S, V] = \text{svd}(\text{Sigma});$$

- From  $[U, S, V] = \text{svd}(\text{Sigma})$  we get;

$$U = \begin{bmatrix} | & | & \dots & | \\ u^{(1)} & u^{(2)} & \dots & u^{(n)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times n}$$

- $\text{Ureduce} = U(:, 1:k);$
- $z = \text{Ureduce}' * x;$

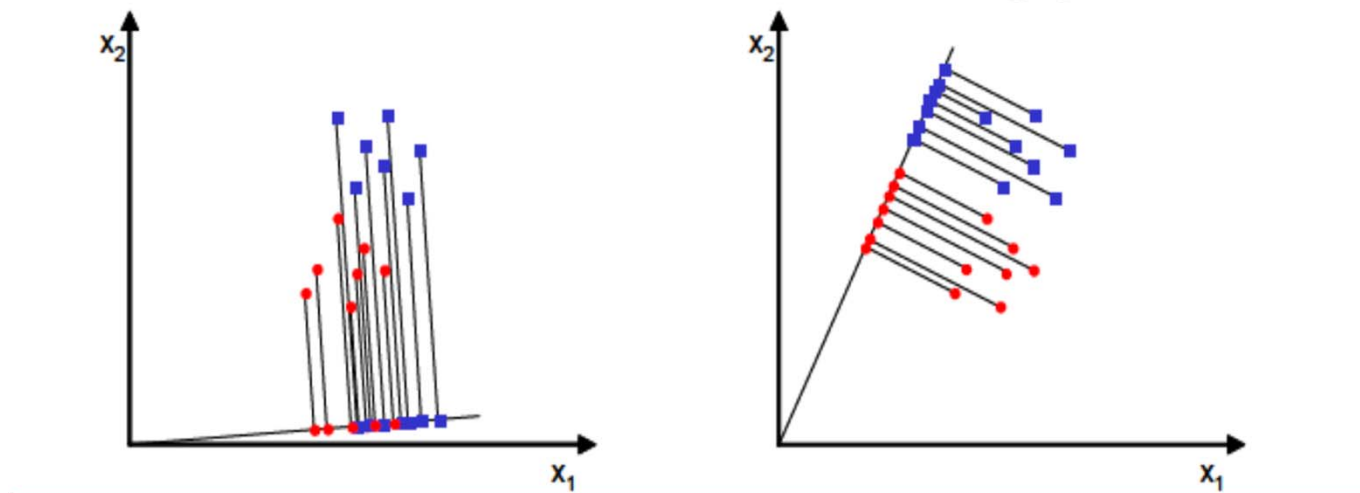
# LDA

---

- Assume we have a set of D-dimensional samples  $\{x(1), x(2), \dots, x(N)\}$ ,  $N_1$  of which belong to class  $\omega_1$ , and  $N_2$  to class  $\omega_2$ . We seek to obtain a scalar  $y$  by projecting the samples  $x$  onto a line

$$y = \mathbf{w}^T \mathbf{x}$$

- Of all the possible lines we would like to select the one that maximizes the separability of the scalars
  - This is illustrated for the two-dimensional case in the following figures



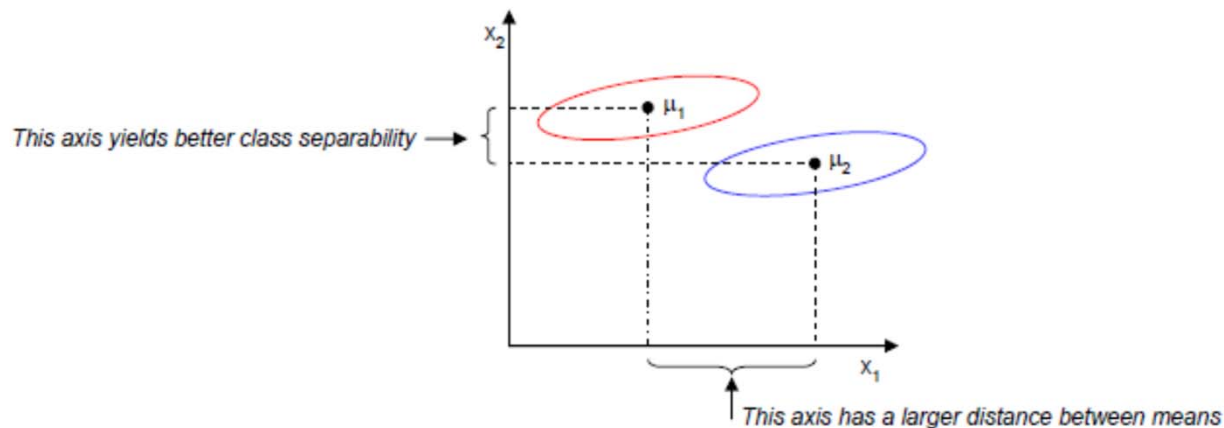
- In order to find a good projection vector, we need to define a measure of separation between the projections
  - The mean vector of each class in  $\mathbf{x}$  and  $\mathbf{y}$  feature space is

$$\mu_i = \frac{1}{N_i} \sum_{\mathbf{x} \in \omega_i} \mathbf{x} \quad \text{and} \quad \tilde{\mu}_i = \frac{1}{N_i} \sum_{\mathbf{y} \in \omega_i} \mathbf{y} = \frac{1}{N_i} \sum_{\mathbf{x} \in \omega_i} \mathbf{w}^T \mathbf{x} = \mathbf{w}^T \mu_i$$

- We could then choose the distance between the projected means as our objective function
- However, the distance between the projected means is not a very good measure since it does not take into account the standard deviation within the classes

$$J(\mathbf{w}) = |\tilde{\mu}_1 - \tilde{\mu}_2| = |\mathbf{w}^T (\mu_1 - \mu_2)|$$

- However, the distance between the projected means is not a very good measure since it does not take into account the standard deviation within the classes



- **The solution proposed by Fisher is to maximize a function that represents the difference between the means, normalized by a measure of the within-class scatter**

- For each class we define the scatter, an equivalent of the variance, as

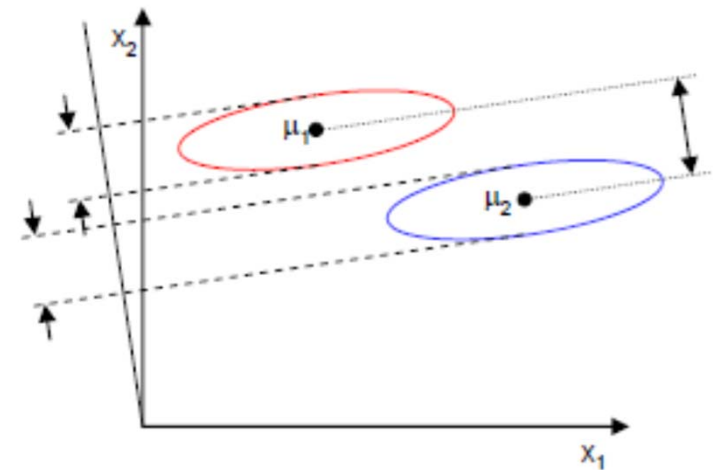
$$\tilde{s}_1^2 = \sum_{y \in \omega_1} (y - \tilde{\mu}_1)^2$$

where the quantity  $(\tilde{s}_1^2 + \tilde{s}_2^2)$  is called the within-class scatter of the projected examples

- The Fisher linear discriminant is defined as the linear function  $\mathbf{w}^T \mathbf{x}$  that maximizes the criterion function

$$J(\mathbf{w}) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- Therefore, we will be looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as far apart as possible



- In order to find the optimum projection  $w^*$ , we need to express  $J(w)$  as an explicit function of  $w$
- We define a measure of the scatter in multivariate feature space  $\mathbf{x}$ , which are scatter matrices

$$S_i = \sum_{\mathbf{x} \in \omega_i} (\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T$$

$$S_1 + S_2 = S_W$$

where  $S_W$  is called the **within-class scatter matrix**

- The scatter of the projection  $\mathbf{y}$  can then be expressed as a function of the scatter matrix in feature space  $\mathbf{x}$

$$\tilde{S}_i^2 = \sum_{\mathbf{y} \in \omega_i} (\mathbf{y} - \tilde{\mu}_i)^2 = \sum_{\mathbf{x} \in \omega_i} (w^T \mathbf{x} - w^T \mu_i)^2 = \sum_{\mathbf{x} \in \omega_i} w^T (\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T w = w^T S_i w$$

$$\tilde{S}_1^2 + \tilde{S}_2^2 = w^T S_W w$$

- Similarly, the difference between the projected means can be expressed in terms of the means in the original feature space

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (w^T \mu_1 - w^T \mu_2)^2 = w^T \underbrace{(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T}_{S_B} w = w^T S_B w$$

The matrix  $S_B$  is called the **between-class scatter**.

- We can finally express the Fisher criterion in terms of  $S_W$  and  $S_B$  as

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

- To find the maximum of  $J(w)$  we derive and equate to zero

$$\begin{aligned}\frac{d}{dw}[J(w)] &= \frac{d}{dw} \left[ \frac{w^T S_B w}{w^T S_W w} \right] = 0 \Rightarrow \\ \Rightarrow [w^T S_W w] \frac{d[w^T S_B w]}{dw} - [w^T S_B w] \frac{d[w^T S_W w]}{dw} &= 0 \Rightarrow \\ \Rightarrow [w^T S_W w] 2S_B w - [w^T S_B w] 2S_W w &= 0\end{aligned}$$

- Dividing by  $w^T S_W w$

$$\begin{aligned}\frac{[w^T S_W w]}{[w^T S_W w]} S_B w - \frac{[w^T S_B w]}{[w^T S_W w]} S_W w &= 0 \Rightarrow \\ \Rightarrow S_B w - J S_W w &= 0 \Rightarrow \\ \Rightarrow S_W^{-1} S_B w - J w &= 0\end{aligned}$$

- Solving the generalized eigenvalue problem ( $S_W^{-1} S_B w = J w$ ) yields

$$w^* = \operatorname{argmax}_w \left\{ \frac{w^T S_B w}{w^T S_W w} \right\} = S_W^{-1} (\mu_1 - \mu_2)$$

- This is known as **Fisher's Linear Discriminant** (1936), although it is not a discriminant but rather a specific choice of direction for the projection of the data down to one dimension



# Bayesian Decision Theory

---

- Design classifiers to make decisions subject to minimizing an expected "risk".
  - The simplest risk is the classification error (i.e., assuming that misclassification costs are equal).
  - When misclassification costs are not equal, the risk can include the cost associated with different misclassifications.

# Terminology

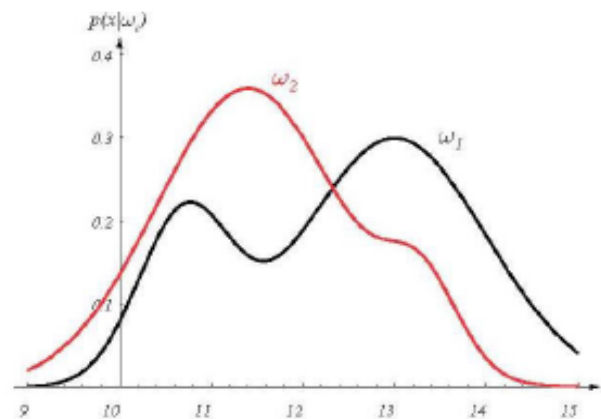
---

- State of nature  $\omega$  (*class label*):
  - e.g.,  $\omega_1$  for sea bass,  $\omega_2$  for salmon
- Probabilities  $P(\omega_1)$  and  $P(\omega_2)$  (*priors*):
  - e.g., prior knowledge of how likely is to get a sea bass or a salmon
- Probability density function  $p(x)$  (*evidence*):
  - e.g., how frequently we will measure a pattern with *feature value  $x$*  (e.g.,  $x$  corresponds to lightness)

## Terminology (cont'd)

- Conditional probability density  $p(x/\omega_j)$  (*likelihood*) :
  - e.g., how frequently we will measure a pattern with **feature value  $x$**  given that the pattern belongs to **class  $\omega_j$**

e.g., lightness distributions  
between salmon/sea-bass  
populations



**FIGURE 2.1.** Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value  $x$  given the pattern is in category  $\omega_i$ . If  $x$  represents the lightness of a fish, the two curves might describe the difference in lightness of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons,

## Terminology (cont'd)

- Conditional probability  $P(\omega_j/x)$  (*posterior*) :
  - e.g., the probability that the fish belongs to class  $\omega_j$  given feature  $x$ .
- Ultimately, we are interested in computing  $P(\omega_j/x)$  for each class  $\omega_j$ .

## Decision Rule Using **Prior** Probabilities Only

**Decide  $\omega_1$  if  $P(\omega_1) > P(\omega_2)$ ; otherwise decide  $\omega_2$**

$$P(\text{error}) = \begin{cases} P(\omega_1) & \text{if we decide } \omega_2 \\ P(\omega_2) & \text{if we decide } \omega_1 \end{cases}$$

**or**  $P(\text{error}) = \min[P(\omega_1), P(\omega_2)]$

- Favours the most likely class.
- This rule will be making the same decision all times.
  - i.e., optimum if no other information is available

# Decision Rule Using **Conditional** Probabilities

- Using **Bayes' rule**:

$$P(\omega_j / x) = \frac{p(x / \omega_j)P(\omega_j)}{p(x)} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

where  $p(x) = \sum_{j=1}^2 p(x / \omega_j)P(\omega_j)$  (i.e., scale factor – sum of probs = 1)

**Decide**  $\omega_1$  if  $P(\omega_1 / x) > P(\omega_2 / x)$ ; otherwise **decide**  $\omega_2$

or

**Decide**  $\omega_1$  if  $p(x/\omega_1)P(\omega_1) > p(x/\omega_2)P(\omega_2)$ ; otherwise **decide**  $\omega_2$

or

**Decide**  $\omega_1$  if  $p(x/\omega_1)/p(x/\omega_2) > P(\omega_2)/P(\omega_1)$ ; otherwise **decide**  $\omega_2$

likelihood ratio

threshold

# Probability of Error

- The **probability of error** is defined as:

$$P(error / x) = \begin{cases} P(\omega_1 / x) & \text{if we decide } \omega_2 \\ P(\omega_2 / x) & \text{if we decide } \omega_1 \end{cases}$$

$$\text{or } P(error/x) = \min[P(\omega_1/x), P(\omega_2/x)]$$

- What is the **average probability error**?

$$P(error) = \int_{-\infty}^{\infty} P(error, x) dx = \int_{-\infty}^{\infty} P(error / x) p(x) dx$$

- The Bayes rule is **optimum**, that is, it minimizes the average probability error!

## A More General Theory

- Use more than one features.
- Allow more than two categories.
- Allow **actions** other than classifying the input to one of the possible categories (e.g., **rejection**).
- Employ a more general error function (i.e., expected “**risk**”) by associating a “**cost**” (based on a “**loss**” function) with different errors.



# Terminology

- Features form a vector  $\mathbf{x} \in R^d$
- A set of  $c$  categories  $\omega_1, \omega_2, \dots, \omega_c$
- A finite set of  $l$  actions  $\alpha_1, \alpha_2, \dots, \alpha_l$
- A *loss* function  $\lambda(\alpha_i / \omega_j)$ 
  - the *cost* associated with taking action  $\alpha_i$  when the correct classification category is  $\omega_j$
- Bayes rule (using vector notation):

$$P(\omega_j / \mathbf{x}) = \frac{p(\mathbf{x} / \omega_j)P(\omega_j)}{p(\mathbf{x})}$$

$$\text{where } p(\mathbf{x}) = \sum_{j=1}^c p(\mathbf{x} / \omega_j)P(\omega_j)$$

## Conditional Risk (or Expected Loss)

- Suppose we observe  $\mathbf{x}$  and take **action**  $\alpha_i$
- The **conditional risk** (or **expected loss**) with taking **action**  $\alpha_i$  is defined as:

$$R(a_i / \mathbf{x}) = \sum_{j=1}^c \lambda(a_i / \omega_j) P(\omega_j / \mathbf{x})$$

# Overall Risk

- Suppose  $\alpha(\mathbf{x})$  is a general decision rule that determines which action  $\alpha_1, \alpha_2, \dots, \alpha_l$  to take for every  $\mathbf{x}$ .
- The overall risk is defined as:

$$R = \int R(\alpha(\mathbf{x}) / \mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

- The optimum decision rule is the *Bayes rule*

## Example: Two-category classification

- Define
  - $\alpha_1$ : decide  $\omega_1$
  - $\alpha_2$ : decide  $\omega_2$
  - $\lambda_{ij} = \lambda(\alpha_i / \omega_j)$

- The conditional risks are:

$$R(a_i / \mathbf{x}) = \sum_{j=1}^c \lambda(a_i / \omega_j) P(\omega_j / \mathbf{x})$$



$$R(a_1 / \mathbf{x}) = \lambda_{11} P(\omega_1 / \mathbf{x}) + \lambda_{12} P(\omega_2 / \mathbf{x})$$

$$R(a_2 / \mathbf{x}) = \lambda_{21} P(\omega_1 / \mathbf{x}) + \lambda_{22} P(\omega_2 / \mathbf{x})$$

## Example: Two-category classification (cont'd)

- Minimum risk decision rule:

**Decide**  $\omega_1$  if  $R(a_1/\mathbf{x}) < R(a_2/\mathbf{x})$ ; otherwise decide  $\omega_2$

or

**Decide**  $\omega_1$  if  $(\lambda_{21} - \lambda_{11})P(\omega_1/\mathbf{x}) > (\lambda_{12} - \lambda_{22})P(\omega_2/\mathbf{x})$ ; otherwise decide  $\omega_2$

or (i.e., using likelihood ratio)

**Decide**  $\omega_1$  if  $\frac{p(\mathbf{x}/\omega_1)}{p(\mathbf{x}/\omega_2)} > \frac{(\lambda_{12} - \lambda_{22})}{(\lambda_{21} - \lambda_{11})} \frac{P(\omega_2)}{P(\omega_1)}$ ; otherwise decide  $\omega_2$

likelihood ratio

threshold

## Special Case: Zero-One Loss Function

- Assign the same loss to all errors:

$$\lambda(a_i/\omega_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases}$$

- The conditional risk corresponding to this loss function:

$$R(a_i/\mathbf{x}) = \sum_{j=1}^c \lambda(a_i/\omega_j)P(\omega_j/\mathbf{x}) = \sum_{i \neq j} P(\omega_j/\mathbf{x}) = 1 - P(\omega_i/\mathbf{x})$$

## Special Case: Zero-One Loss Function (cont'd)

- The decision rule becomes:

**Decide  $\omega_1$**  if  $R(a_1/\mathbf{x}) < R(a_2/\mathbf{x})$ ; otherwise decide  $\omega_2$

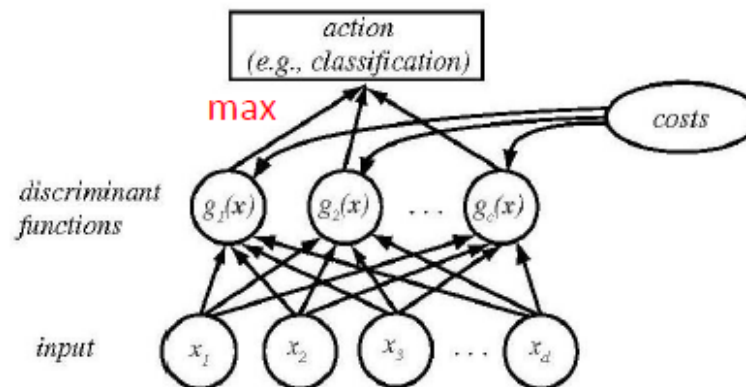
**or** **Decide  $\omega_1$**  if  $1 - P(\omega_1/\mathbf{x}) < 1 - P(\omega_2/\mathbf{x})$ ; otherwise decide  $\omega_2$

**or** **Decide  $\omega_1$**  if  $P(\omega_1/\mathbf{x}) > P(\omega_2/\mathbf{x})$ ; otherwise decide  $\omega_2$

# Discriminant Functions

- A useful way to represent a classifier is through **discriminant functions**  $g_i(\mathbf{x})$ ,  $i = 1, \dots, c$ , where a feature vector  $\mathbf{x}$  is assigned to class  $\omega_i$  if:

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \text{ for all } j \neq i$$





# Discriminants for Bayes Classifier

- Assuming a **general loss** function:

$$g_i(\mathbf{x}) = -R(\alpha_i / \mathbf{x})$$

- Assuming the **zero-one loss** function:

$$g_i(\mathbf{x}) = P(\omega_i / \mathbf{x})$$

# Discriminants for Bayes Classifier (cont'd)

- Is the choice of  $g_i$  unique?
  - Replacing  $g_i(\mathbf{x})$  with  $f(g_i(\mathbf{x}))$ , where  $f()$  is **monotonically increasing**, does not change the classification results.

$$g_i(\mathbf{x}) = P(\omega_i / \mathbf{x}) \quad \Rightarrow \quad \begin{aligned} g_i(\mathbf{x}) &= \frac{p(\mathbf{x} / \omega_i) P(\omega_i)}{p(\mathbf{x})} \\ g_i(\mathbf{x}) &= p(\mathbf{x} / \omega_i) P(\omega_i) \\ g_i(\mathbf{x}) &= \ln p(\mathbf{x} / \omega_i) + \ln P(\omega_i) \end{aligned}$$

we'll use this  
discriminant extensively!

# Case of two categories

- More common to use a single discriminant function (*dichotomizer*) instead of two:

$$g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x})$$

**Decide**  $\omega_1$  if  $g(\mathbf{x}) > 0$ ; otherwise decide  $\omega_2$

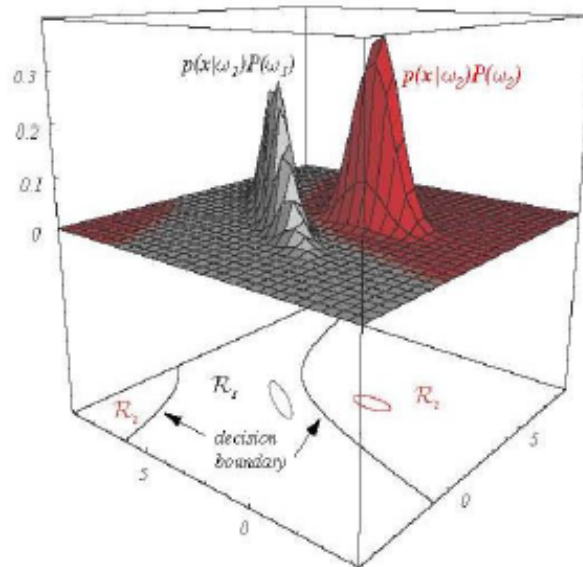
- Examples:

$$g(\mathbf{x}) = P(\omega_1 / \mathbf{x}) - P(\omega_2 / \mathbf{x})$$

$$g(\mathbf{x}) = \ln \frac{p(\mathbf{x} / \omega_1)}{p(\mathbf{x} / \omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

# Decision Regions and Boundaries

- Discriminants divide the feature space in *decision regions*  $R_1, R_2, \dots, R_c$ , separated by *decision boundaries*.



Decision boundary  
is defined by:

$$g_1(\mathbf{x}) = g_2(\mathbf{x})$$

## Discriminant Function for Multivariate Gaussian Density

$$N(\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \mu)^t \Sigma^{-1} (\mathbf{x} - \mu)\right]$$

- Consider the following discriminant function:

$$g_i(\mathbf{x}) = \ln p(\mathbf{x} / \omega_i) + \ln P(\omega_i)$$

- If  $p(\mathbf{x} / \omega_i) \sim N(\mu_i, \Sigma_i)$ , then

$$g_i(\mathbf{x}) = -\frac{1}{2} (\mathbf{x} - \mu_i)^t \Sigma_i^{-1} (\mathbf{x} - \mu_i) - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

# Multivariate Gaussian Density: **Case I**

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^t \Sigma_i^{-1}(\mathbf{x} - \mu_i) - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

- $\Sigma_i = \sigma^2 \mathbf{I}$  (diagonal matrix)
  - Features are statistically independent
  - Each feature has the same variance

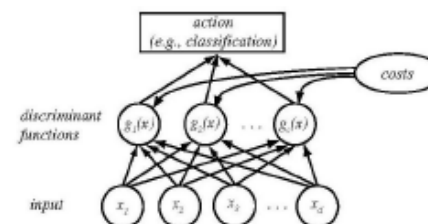
- If we disregard  $\frac{d}{2} \ln 2\pi$  and  $\frac{1}{2} \ln |\Sigma_i|$  (constants):

$$g_i(\mathbf{x}) = -\frac{\|\mathbf{x} - \mu_i\|^2}{2\sigma^2} + \ln P(\omega_i)$$

where  $\|\mathbf{x} - \mu_i\|^2 = (\mathbf{x} - \mu_i)^t(\mathbf{x} - \mu_i)$

- Expanding the above expression:

$$g_i(\mathbf{x}) = -\frac{1}{2\sigma^2} [\mathbf{x}^t \mathbf{x} - 2\mu_i^t \mathbf{x} + \mu_i^t \mu_i] + \ln P(\omega_i)$$



# Multivariate Gaussian Density: Case I (cont'd)

- Disregarding  $\mathbf{x}^t \mathbf{x}$  (constant), we get a linear discriminant:

$$g_i(\mathbf{x}) = \mathbf{w}_i^t \mathbf{x} + w_{i0}$$

where  $\mathbf{w}_i = \frac{1}{\sigma^2} \mu_i$ , and  $w_{i0} = -\frac{1}{2\sigma^2} \mu_i^t \mu_i + \ln P(\omega_i)$

- Decision boundary is determined by hyperplanes; setting  $g_i(\mathbf{x}) = g_j(\mathbf{x})$ :

$$\mathbf{w}^t (\mathbf{x} - \mathbf{x}_0) = 0$$

where  $\mathbf{w} = \mu_i - \mu_j$ , and  $\mathbf{x}_0 = \frac{1}{2} (\mu_i + \mu_j) - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \frac{P(\omega_i)}{P(\omega_j)} (\mu_i - \mu_j)$

## Multivariate Gaussian Density: **Case II**

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^t \Sigma_i^{-1}(\mathbf{x} - \mu_i) - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

- $\Sigma_i = \Sigma$

- The clusters have hyperellipsoidal shape and same size (centered at  $\mu$ ).

- If we disregard  $\frac{d}{2} \ln 2\pi$  and  $\frac{1}{2} \ln |\Sigma_i|$  (constants):

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^t \Sigma^{-1}(\mathbf{x} - \mu_i) + \ln P(\omega_i)$$

- Expanding the above expression and disregarding the quadratic term:

$$g_i(\mathbf{x}) = \mathbf{w}_i^t \mathbf{x} + w_{i0}$$

(linear discriminant)

where  $\mathbf{w}_i = \Sigma^{-1} \mu_i$ , and  $w_{i0} = -\frac{1}{2} \mu_i^t \Sigma^{-1} \mu_i + \ln P(\omega_i)$



## Multivariate Gaussian Density: Case II (cont'd)

- Decision boundary is determined by hyperplanes; setting  $g_i(\mathbf{x}) = g_j(\mathbf{x})$ :

$$\mathbf{w}^t(\mathbf{x} - \mathbf{x}_0) = 0$$

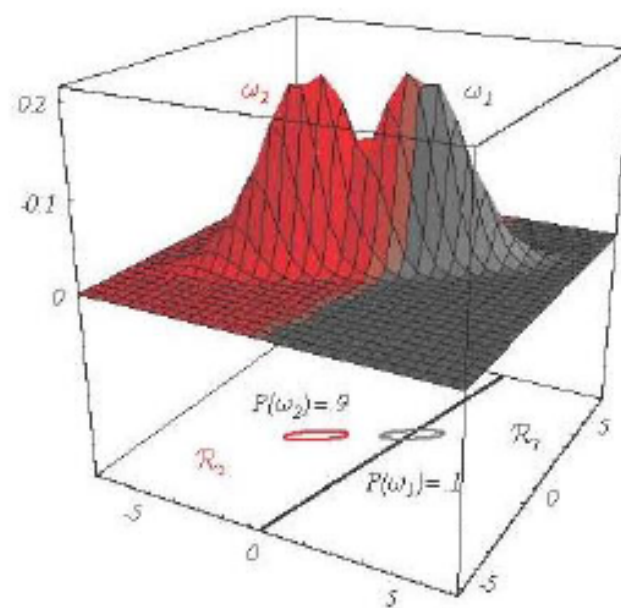
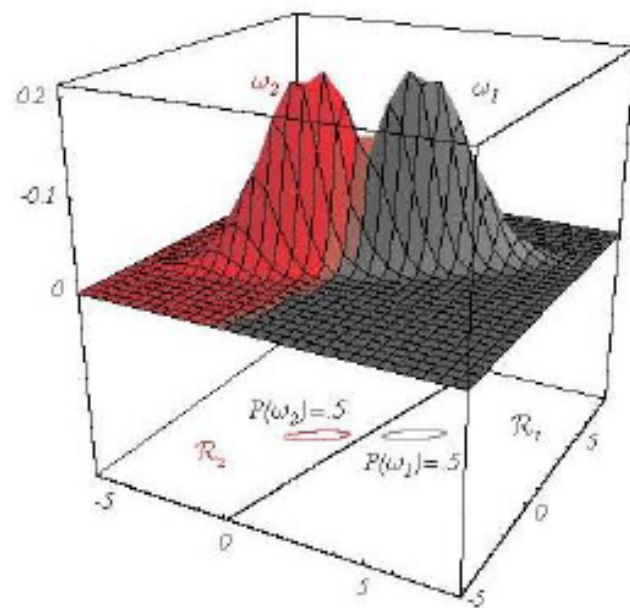
where  $\mathbf{w} = \Sigma^{-1}(\mu_i - \mu_j)$  and  $\mathbf{x}_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln[P(\omega_i)/P(\omega_j)]}{(\mu_i - \mu_j)^t \Sigma^{-1}(\mu_i - \mu_j)} (\mu_i - \mu_j)$

- **Mahalanobis distance classifier**
  - When  $P(\omega_i)$  are equal, then the discriminant becomes:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^t \Sigma^{-1}(\mathbf{x} - \mu_i) + \ln P(\omega_i)$$



$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^t \Sigma^{-1}(\mathbf{x} - \mu_i)$$



If  $P(\omega_i) \neq P(\omega_j)$ , then  $\mathbf{x}_0$  shifts away from the most likely category.

## Multivariate Gaussian Density: Case III

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^t \Sigma_i^{-1}(\mathbf{x} - \mu_i) - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

- $\Sigma_i$  = arbitrary

- The clusters have different shapes and sizes (centered at  $\mu$ ).

- If we disregard  $\frac{d}{2} \ln 2\pi$  (constant):

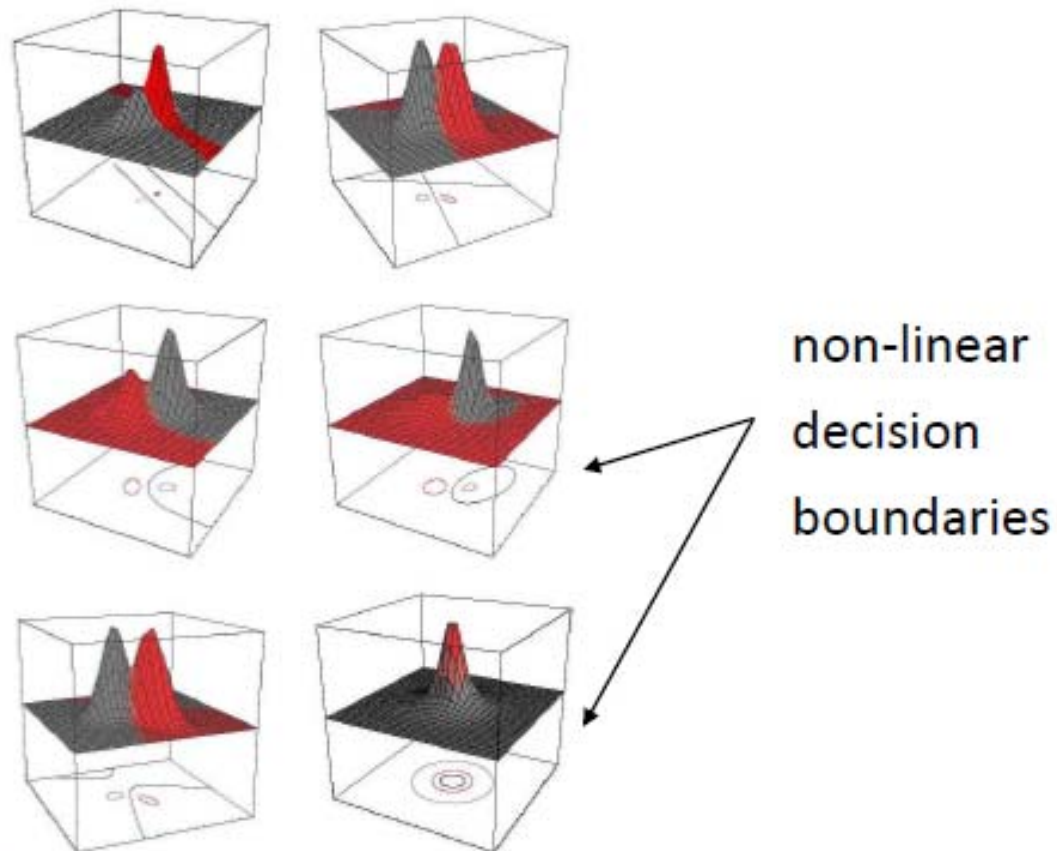
$$g_i(\mathbf{x}) = \mathbf{x}^t \mathbf{W}_i \mathbf{x} + \mathbf{w}_i^t \mathbf{x} + w_{i0}$$

(quadratic discriminant)

where  $\mathbf{W}_i = -\frac{1}{2} \Sigma_i^{-1}$ ,  $\mathbf{w}_i = \Sigma_i^{-1} \mu_i$ , and  $w_{i0} = -\frac{1}{2} \mu_i^t \Sigma_i^{-1} \mu_i - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$

- Decision boundary is determined by hyperquadrics; setting  $g_i(\mathbf{x}) = g_j(\mathbf{x})$

e.g., hyperplanes, pairs of hyperplanes, hyperspheres, hyperellipsoids, hyperparaboloids etc.



**FIGURE 2.14.** Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadric, one can find two Gaussian distributions whose Bayes decision boundary is that hyperquadric. These variances are indicated by the contours of constant probability density. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

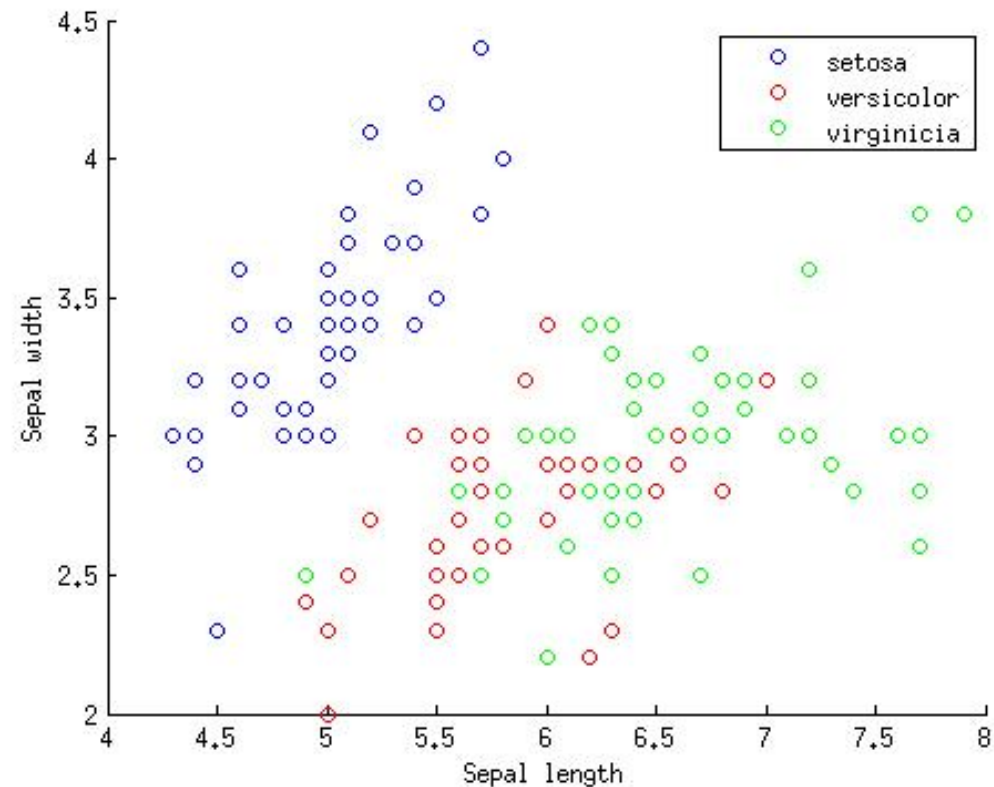
# Classification: Example Iris Dataset

---

- 3 classes, 4 numeric attributes, 150 instances
- A data set with 150 points and 3 classes. Each point is a random sample of measurements of flowers from one of three iris species - setosa, versicolor, and virginica - collected by Anderson (1935). Used by Fisher (1936) for linear discriminant function technique.
- The measurements are sepal length, sepal width, petal length, and petal width in cm.

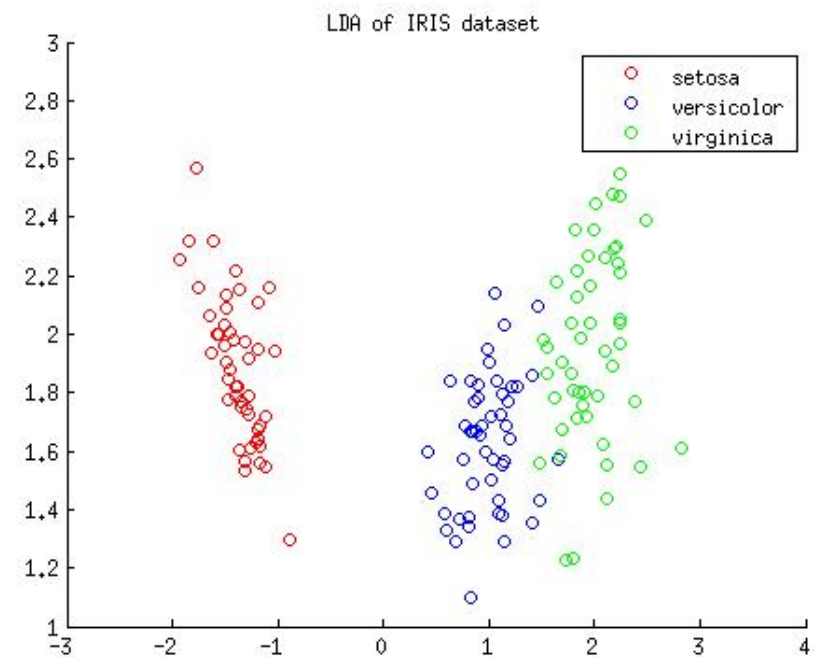
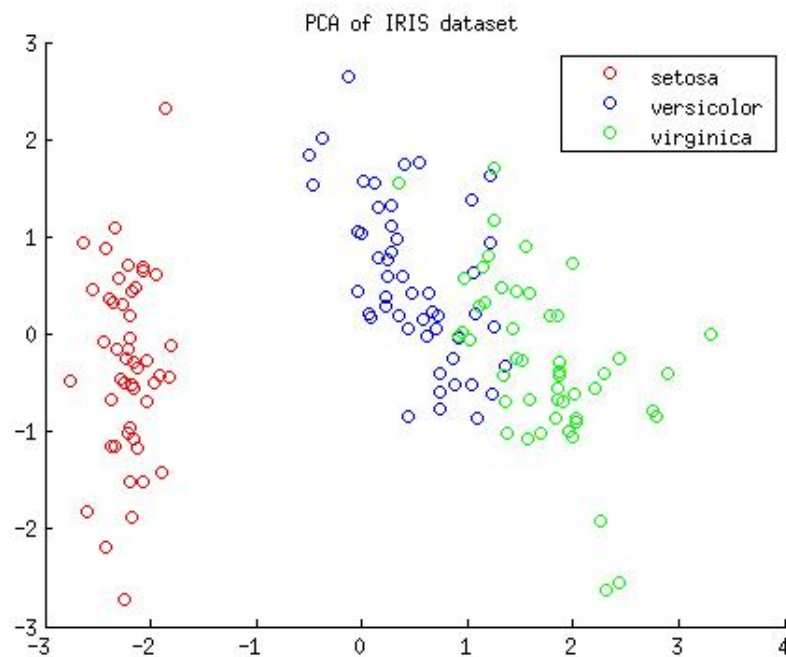
## Data:

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
...
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
...
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
7.1,3.0,5.9,2.1,Iris-virginica
```



# Feature Extraction: PCA and LDA

---



# Results: Accuracy in Classification


---

CLASSIFIERS DATA	LINEAR $\Sigma(i) = \sigma^2 I$	LINEAR $\Sigma(i) = \Sigma$	QUADRATIC $\Sigma(i) = \text{arbitrary}$
ORIGINAL DATA	94.86%	97.14%	98.0%
TRANSFORMED DATA(USING LDA)	98.0%	97.14%	98.0%
TRANSFORMED DATA(USING PCA)	84.2%	88.14%	90.0%



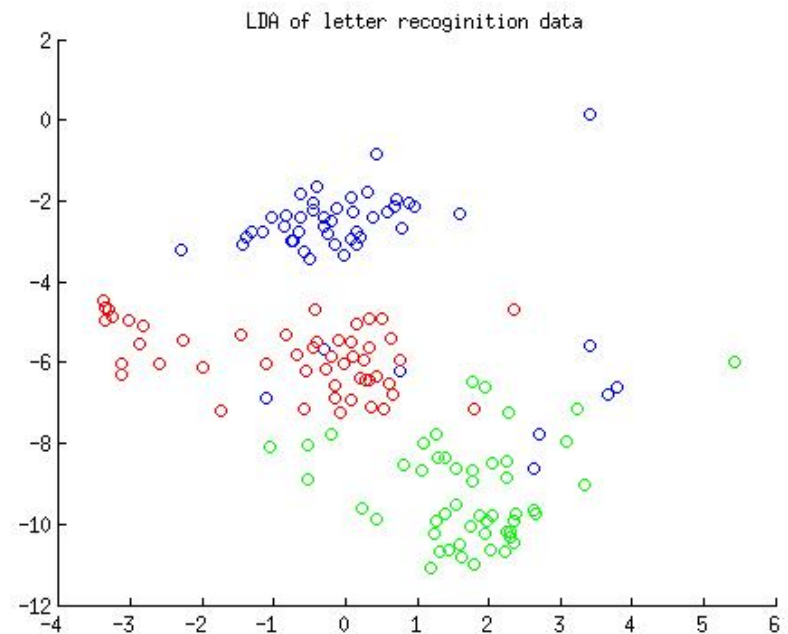
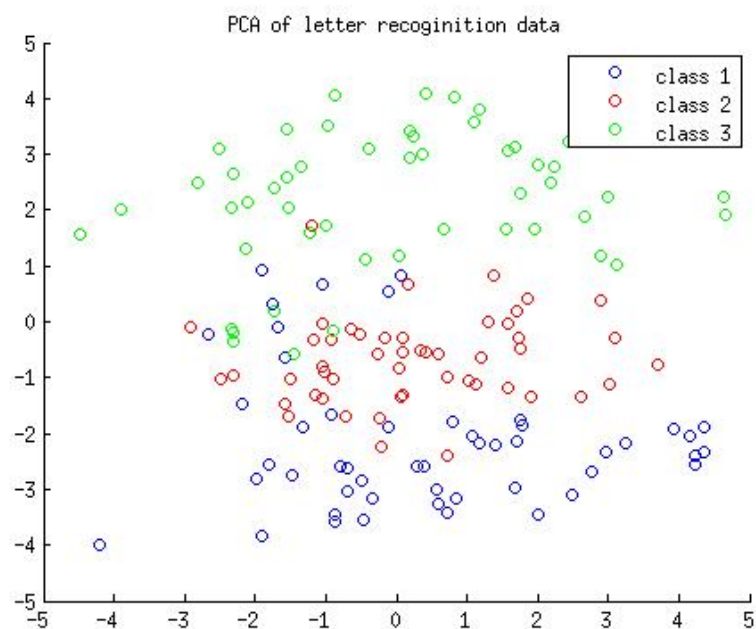
# Classification: Example Letter Recognition Dataset

---

- 26 classes, 16 numeric attributes, 5000 instances
  - A data set with 5000 points and 26 classes follows multivariate normal distribution
  - Each class represents a particular alphabetical letter like A, B, C, F, Z etc.
- 
- A solid green horizontal bar spanning the width of the slide, located at the bottom.

# Feature Extraction: PCA and LDA

---

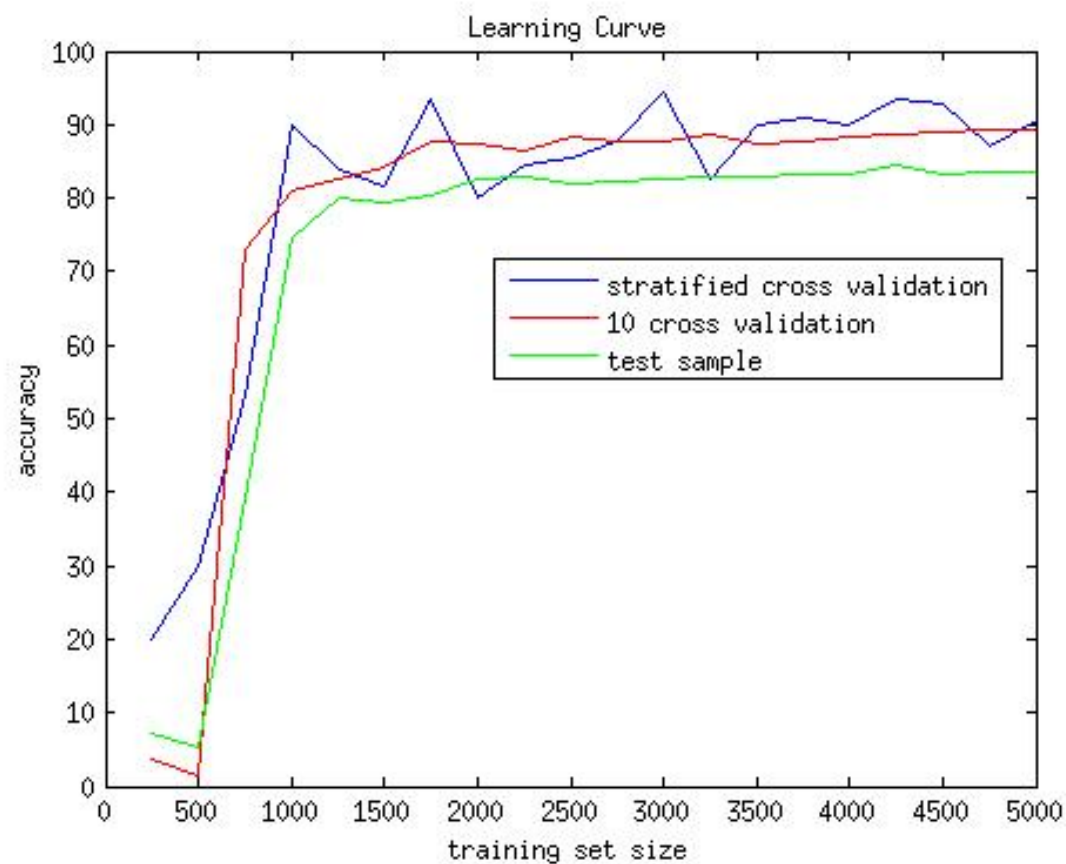


# Results: Accuracy in Classification

---

CLASSIFIERS DATA	LINEAR $\Sigma(i) = \sigma^2 I$	LINEAR $\Sigma(i) = \Sigma$	QUADRATIC $\Sigma(i) = \text{arbitrary}$
ORIGINAL DATA	58.3%	85.88%	92.5%
TRANSFORMED DATA(USING LDA)	40.41%	76.47%	92.35%
TRANSFORMED DATA(USING PCA)	37.65%	71.14%	90.76%

# Results

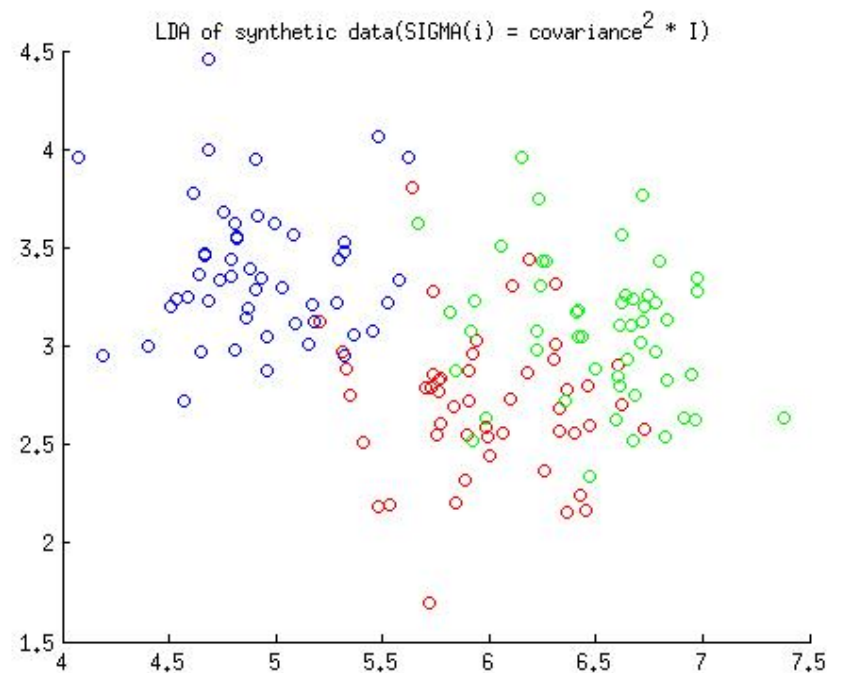
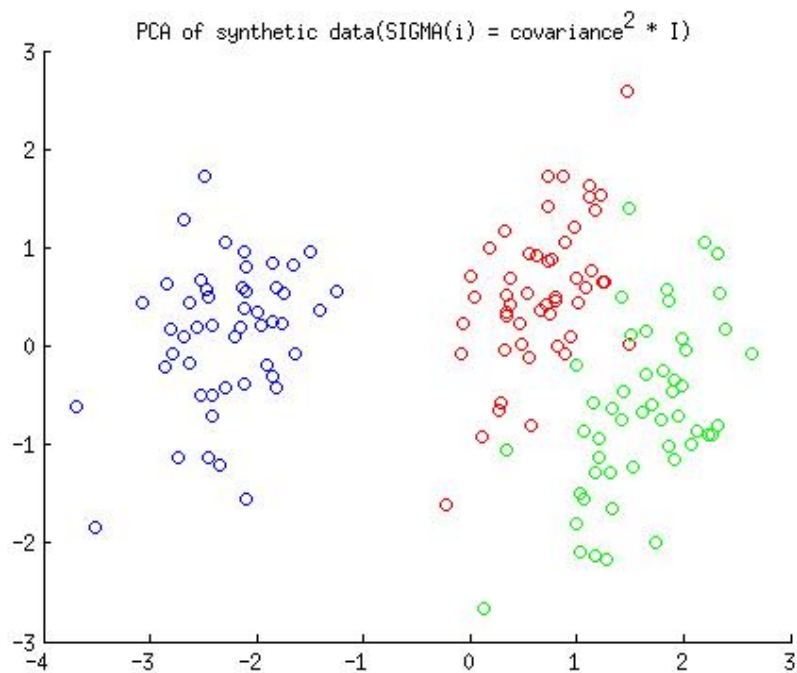


# Classification: Example Synthetic Dataset

---

- 3 classes, 5 numeric attributes, 5000 instances.
- A data set with 5000 points and 3 classes follows multivariate normal distribution.
- Each class has same diagonal covariance matrices with different means.

# Feature Extraction: PCA and LDA



# Results: Accuracy in Classification

---

CLASSIFIERS DATA	LINEAR $\Sigma(i) = \sigma^2 I$	LINEAR $\Sigma(i) = \Sigma$	QUADRATIC $\Sigma(i) = \text{arbitrary}$
ORIGINAL DATA	98.26%	98.56%	98.32%
TRANSFORMED DATA(USING LDA)	99.04%	98.56%	98.50%
TRANSFORMED DATA(USING PCA)	94.07%	96.4%	95.45%