Statistical pattern recognition

Bayes theorem

Problem: deciding if a patient has a particular condition based on a particular test

- However, the test is *imperfect*
  - Someone with the condition may go undetected (false negative)
  - Someone without condition may come out positive (false positive)
- Test properties
  - SPECIFICITY or true-negative rate P(NEG|¬COND)
  - SENSITIVITY or true-positive rate P(POS|COND)

Problem definition

- Assume a population of **10,000** where **1** out of every **100** people has the medical condition
- Assume that we design a test with **98%** specificity *P(NEG|¬COND)* and **90%** sensitivity *P(POS|COND)*

- You take the test, and it comes POSITIVE

- What conditional probability are we after?
- How likely is it that you have the condition?

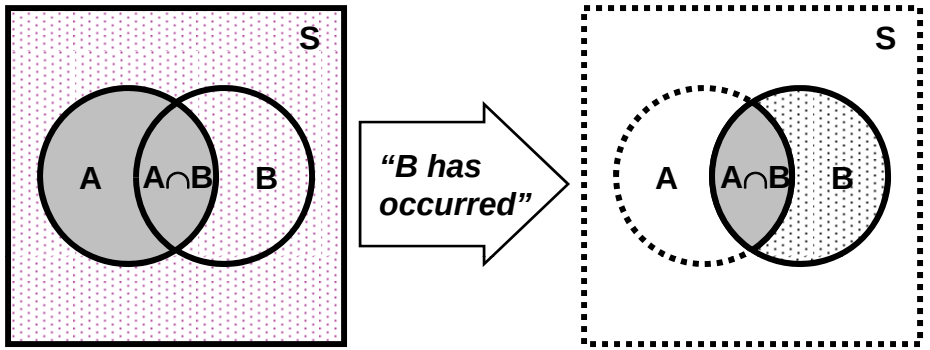
Solution: Joint frequency table

– The answer is the ratio of individuals with the condition to total individuals (considering only individuals that tested positive) or 90/288=0.3125

	TEST IS POSITIVE	TEST IS NEGATIVE	ROW TOTAL
HAS CONDITION	*True-positive* *P(POS|COND)* **100×0.90=90**	*False-negative* *P(NEG|COND)* **100×(1-0.90)=10**	**100**
FREE OF CONDITION	*False-positive* *P(POS|¬COND)* **9,900×(1-0.98)=198**	*True-negative* *P(NEG|¬COND)* **9,900×0.98=9,072**	**9,900**
COLUMN TOTAL	**288**	**9,712**	**10,000**

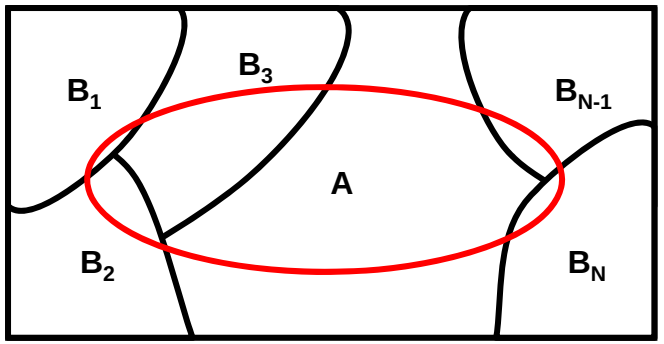
Conditional probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{for} \quad P(B) > 0$$



Total probability

$$\begin{aligned} P(A) &= P(A \cap S) = \\ &= P(A \cap B_1) + \ldots + P(A \cap B_N) = \\ &= P(A|B_1)P(B_1) + \ldots + P(A|B_N)P(B_N) = \\ &= \sum_{k=1}^{N} P(A|B_k)P(B_k) \end{aligned}$$



Alternative solution: Bayes theorem

$$P(A|B) = \frac{P(B \mid A)P(A)}{P(B)}$$

$$P(\text{cond}|+) = \frac{P(+|\text{cond}) \cdot P(\text{cond})}{P(+)} =$$

$$= \frac{P(+|\text{cond}) \cdot P(\text{cond})}{P(+|\text{cond}) \cdot P(\text{cond}) + P(+|\neg\text{cond}) \cdot P(\neg\text{cond})} =$$

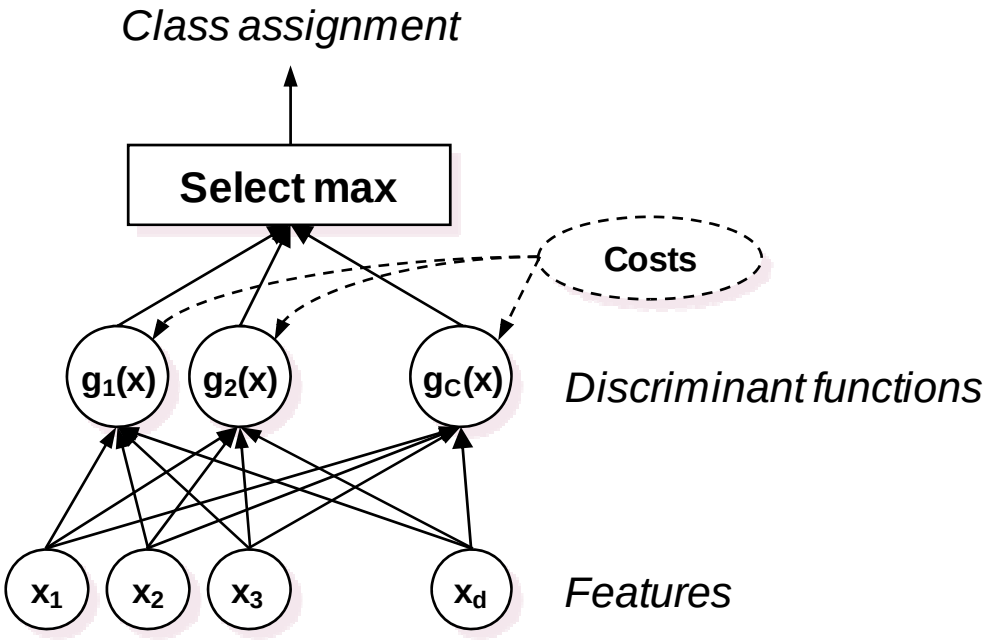
$$= \frac{0.90 \times 0.01}{0.90 \times 0.01 + (1 - 0.98) \times 0.99} = 0.3125$$

In SPR, Bayes theorem is expressed as

$$\text{Posterior}\quad P(\omega_j|x) = \frac{P(x|\omega_j)\cdot P(\omega_j)}{\sum_{k=1}^{N} P(x|\omega_k)\cdot P(\omega_k)} = \frac{\overbrace{P(x|\omega_j)}^{\text{Likelihood}}\cdot \overbrace{P(\omega_j)}^{\text{Prior}}}{\underbrace{P(x)}_{\text{Norm constant}}}$$

- And we assign sample *x* to the class ω_k with the highest posterior
  - It can be shown this rule minimizes the prob. of error

Discriminant functions



$$x \in \omega_i \Leftrightarrow g_i(x) > g_j(x) \ \forall j \neq i$$

$$\text{where } g_i(x) = p(\omega_i \mid x)$$

Quadratic classifiers

For normally distributed classes, the posterior can be reduced to a very simple expression

- Recall an n-dimensional Gaussian density is

$$p(x) = \frac{1}{(2\pi)^{n/2}\left|\sum\right|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^T \sum^{-1} (x-\mu)\right]$$

- Using Bayes rule, the DF can be written as

$$g_i(x) = P(\omega_i \mid x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)} =$$

$$= \frac{1}{(2\pi)^{n/2}\left|\sum_i\right|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_i)^T \sum_i^{-1} (x-\mu_i)\right] P(\omega_i) \frac{1}{P(x)}$$

– Eliminating constant terms

$$g_i(x) = \left|\sum_i\right|^{-1/2} \exp\left[-\frac{1}{2}(x-\mu_i)^T \sum_i^{-1}(x-\mu_i)\right] P(\omega_i)$$

– And taking logs

$$\boxed{g_i(x) = -\frac{1}{2}(x-\mu_i)^T \sum_i^{-1}(x-\mu_i) - \frac{1}{2}\log\left|\sum_i\right| + \log P(\omega_i)}$$

– This is known as a quadratic discriminant function (because it is a function of x^2)

Case 1: $\Sigma_i=\sigma^2 I$

Features are statistically independent, and have the same variance for all classes

– In this case, the quadratic discriminant function becomes

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T\left(\sigma^2 I\right)^{-1}(x-\mu_i) - \frac{1}{2}\log\left|\sigma^2 I\right| + \log P(\omega_i)$$

$$= -\frac{1}{2\sigma^2}(x-\mu_i)^T(x-\mu_i) + \log P(\omega_i)$$

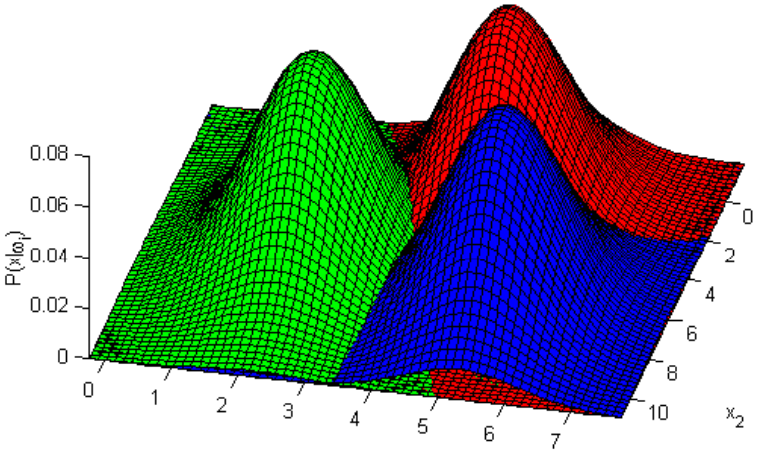
– Assuming equal priors and dropping constant terms

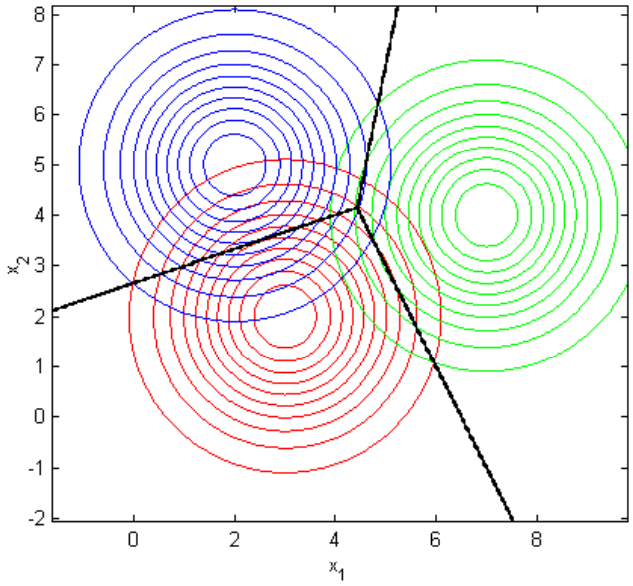
$$g_i(x) = -(x-\mu_i)^T(x-\mu_i) = -\sum_{i=1}^{DIM}\left(x_i-\mu_i\right)^2$$

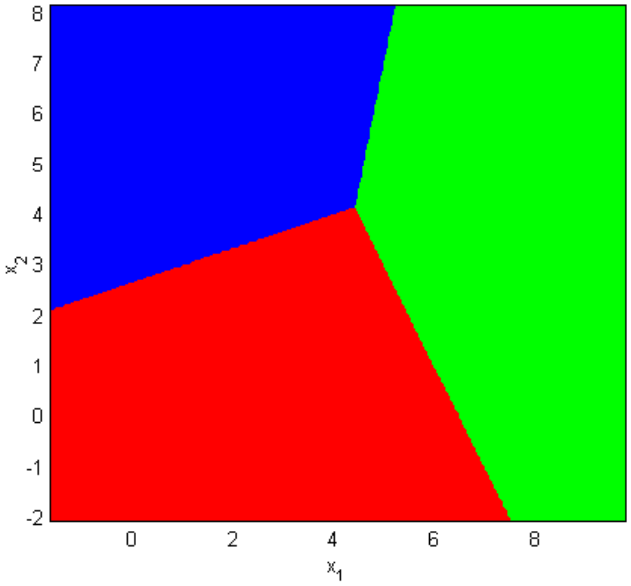
– This is called an Euclidean-distance or nearest mean classifier

$$\mu_1 = \begin{bmatrix} 3 & 2 \end{bmatrix}^T \quad \mu_2 = \begin{bmatrix} 7 & 4 \end{bmatrix}^T \quad \mu_3 = \begin{bmatrix} 2 & 5 \end{bmatrix}^T$$

$$\Sigma_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$







Case 2: $\Sigma_i=\Sigma$

All classes have the same covariance matrix, but the matrix is not diagonal

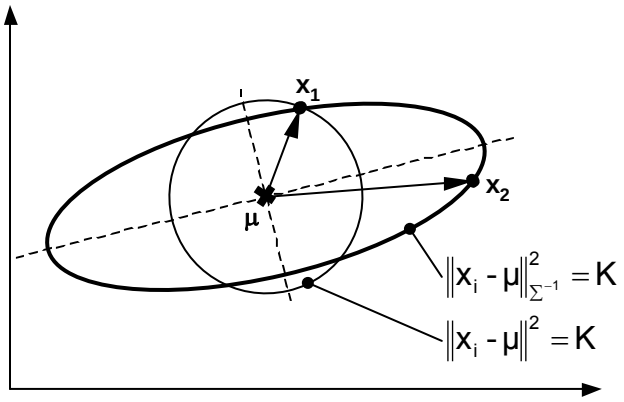
- In this case, the quadratic discriminant becomes

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma^{-1} (x-\mu_i) - \frac{1}{2}\log(|\Sigma|) + \log(P(\omega_i))$$

- assuming equal priors and eliminating constants

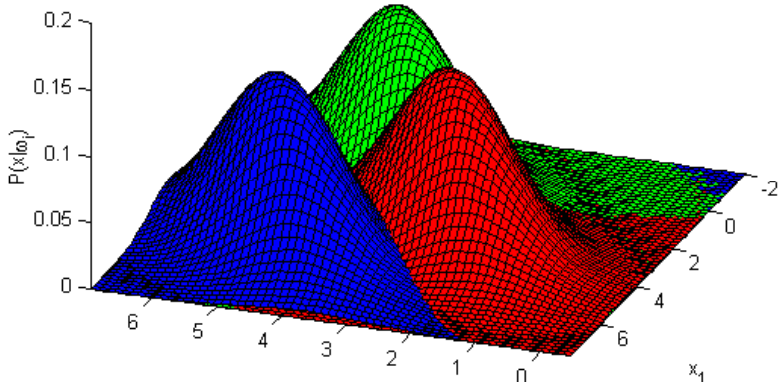
$$\boxed{g_i(x) = -(x-\mu_i)^T \Sigma^{-1} (x-\mu_i)}$$

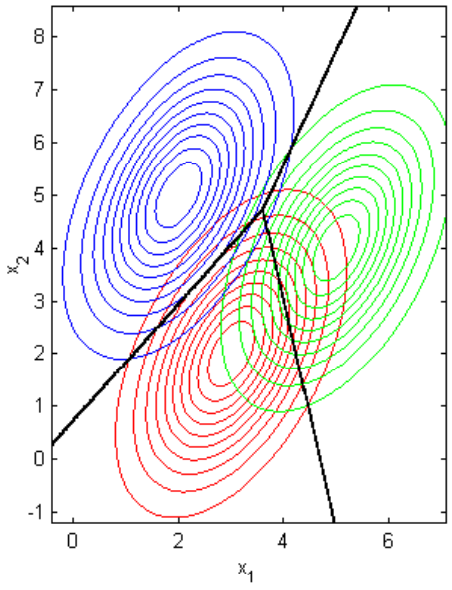
- This is known as a Mahalanobis-distance classifier

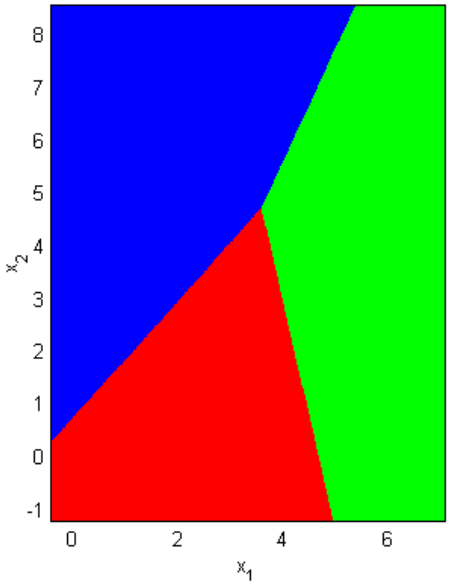


$$\mu_1 = [3 \quad 2]^T \qquad \mu_2 = [5 \quad 4]^T \qquad \mu_3 = [2 \quad 5]^T$$

$$\Sigma_1 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 2 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 2 \end{bmatrix} \qquad \Sigma_3 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 2 \end{bmatrix}$$





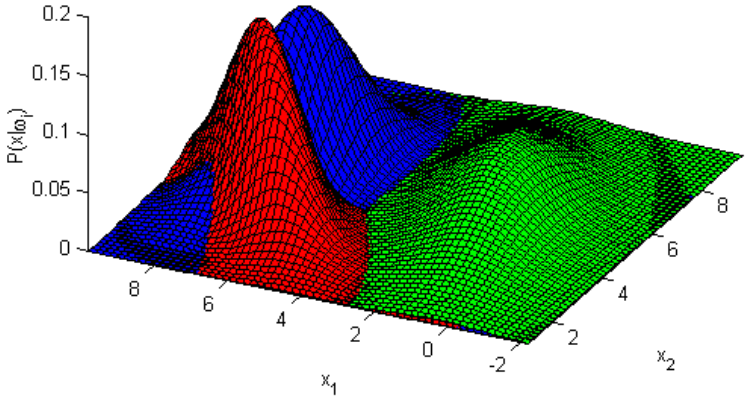


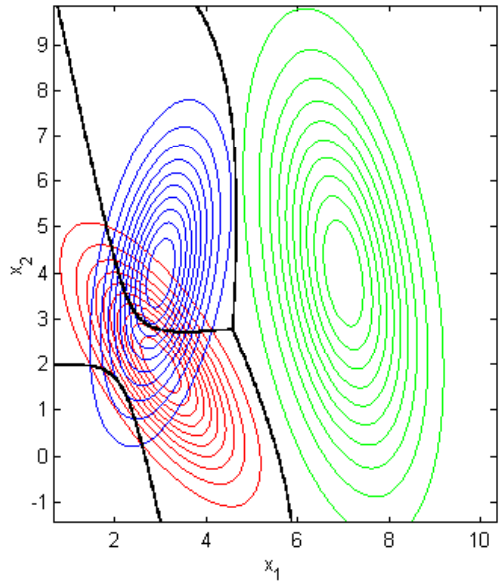
General case

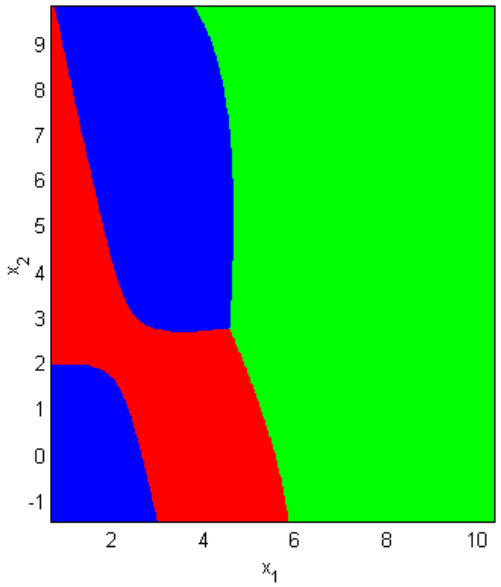
$$\mu_1 = [3 \quad 2]^T \qquad \Sigma_1 = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$

$$\mu_2 = [5 \quad 4]^T \qquad \Sigma_2 = \begin{bmatrix} 1 & -1 \\ -1 & 7 \end{bmatrix}$$

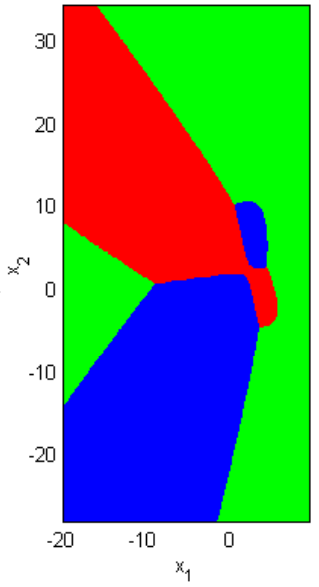
$$\mu_3 = [2 \quad 5]^T \qquad \Sigma_3 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 3 \end{bmatrix}$$







Zoom out

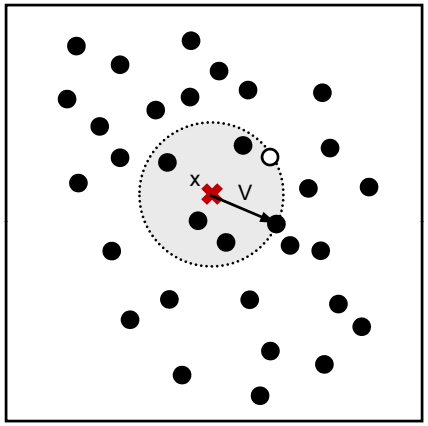


k nearest neighbors

Non-parametric approximation

– Likelihood of each class

$$P(x|\omega_i) = \frac{k_i}{N_i V}$$



– And priors

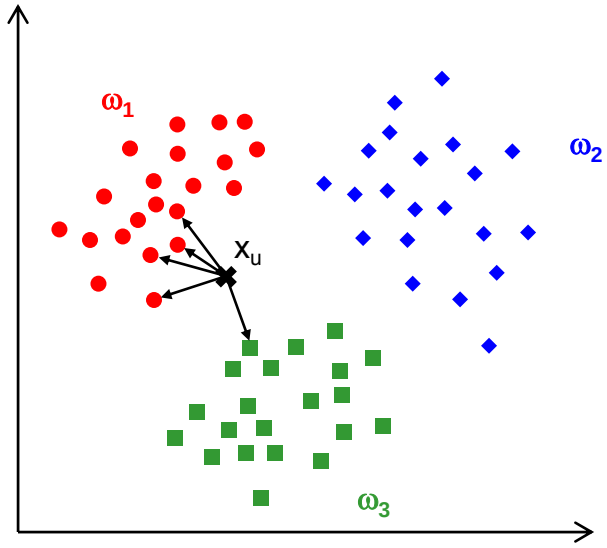
$$P(\omega_i) = \frac{N_i}{N}$$

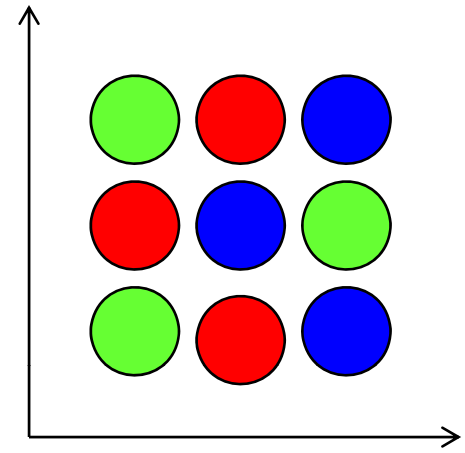
– Then, the posterior becomes

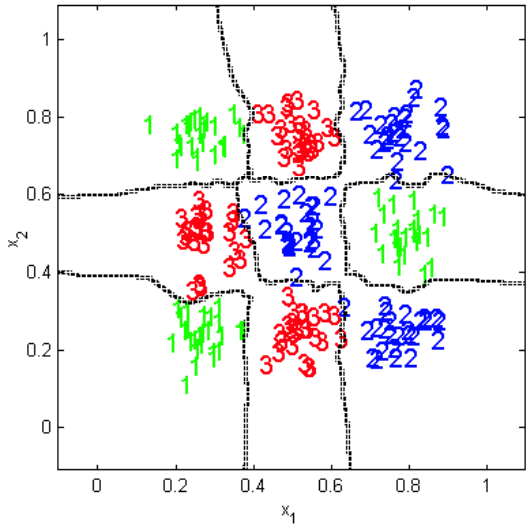
$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)} = \frac{\frac{k_i}{N_i V} \cdot \frac{N_i}{N}}{\frac{k}{NV}} = \frac{k_i}{k}$$

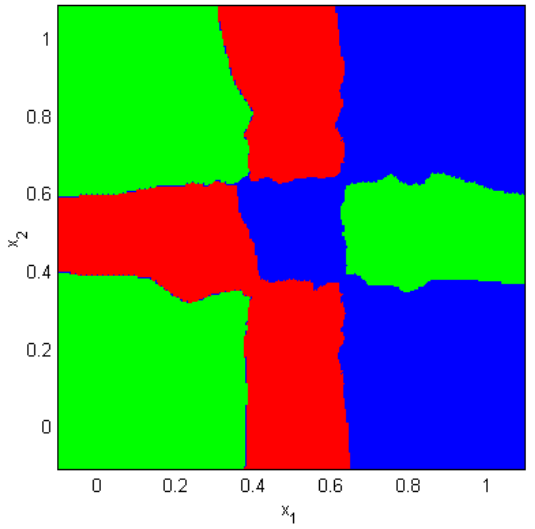
Example

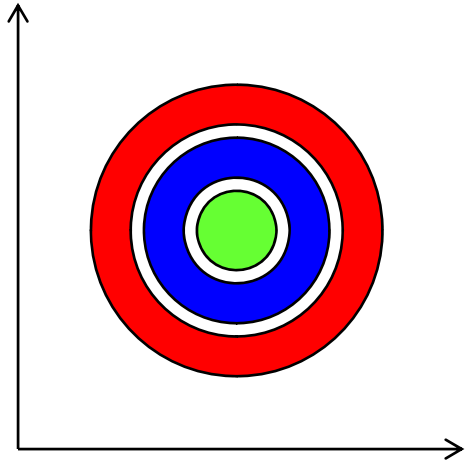
- Given the three classes, assign a class label for the unknown example x_u
- Assume the Euclidean distance and k=5 neighbors
- Of the 5 closest neighbors, 4 belong to ω_1 and 1 belongs to ω_3, so x_u is assigned to ω_1, the predominant class

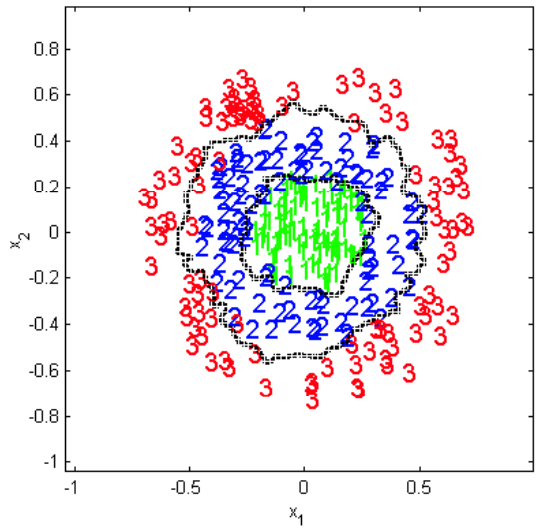


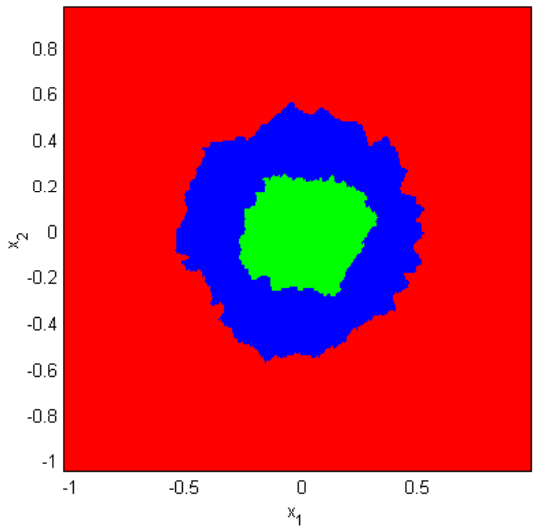



1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
x2
x1

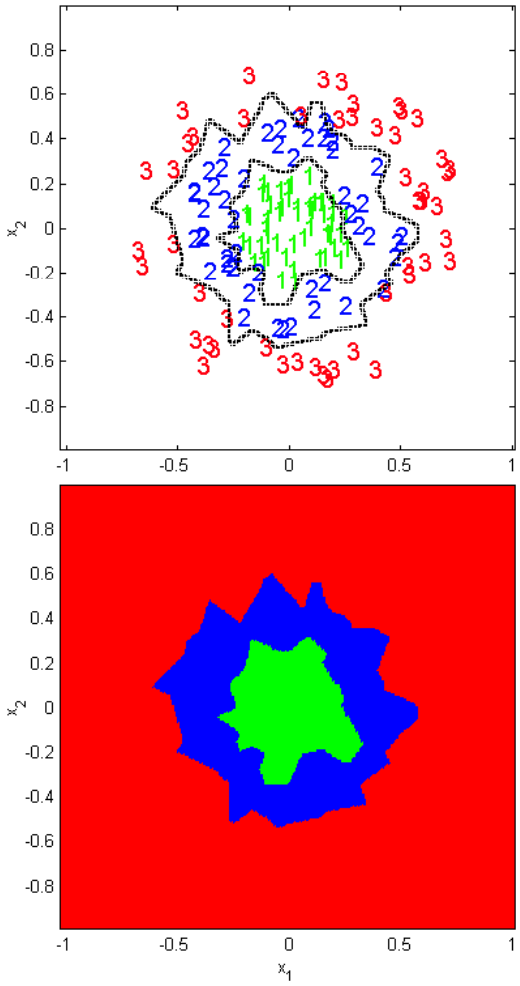

1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
x2
x1



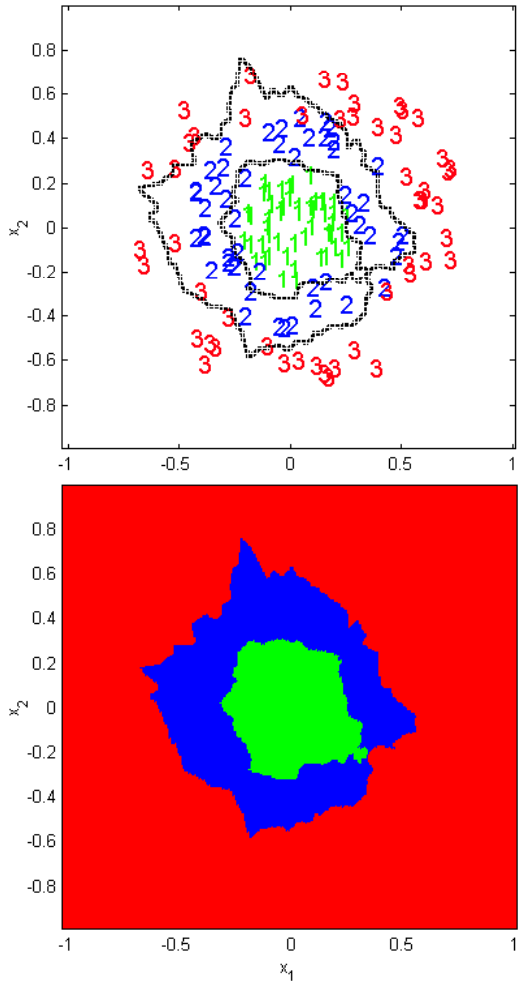

0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
x2
x1


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
x2
x1

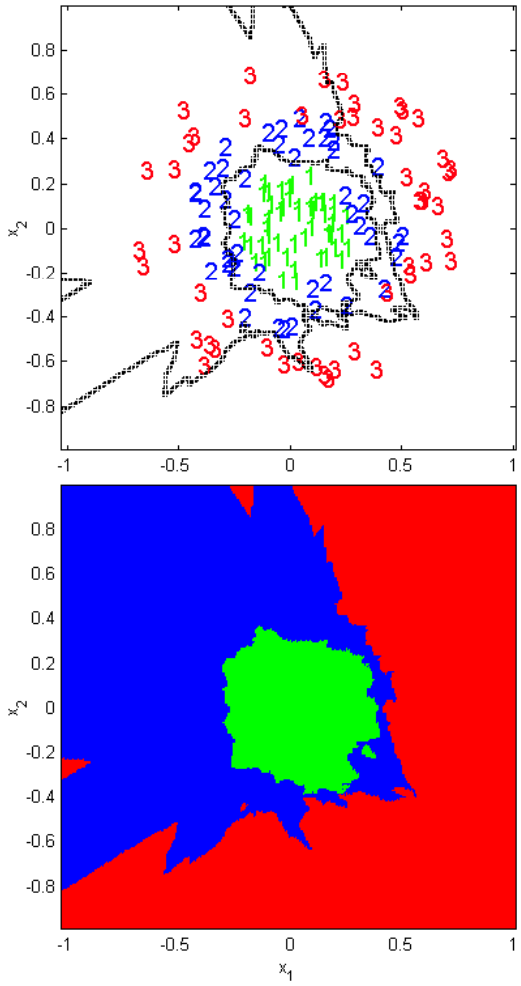
1-NN


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1
x2
x1

5-NN


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1
x2
x1

20-NN


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1
x2
x1

Advantages

- Simple implementation
- Nearly optimal in the large sample limit ($N \rightarrow \infty$)

$$P[error]_{Bayes} < P[error]_{1NN} < 2P[error]_{Bayes}$$

- Uses local information, which can yield highly adaptive behavior
- Lends itself very easily to parallel implementations

Disadvantages

- Large storage requirements
- Computationally intensive recall
- Highly susceptible to the curse of dimensionality

Dimensionality reduction

Why do dimensionality reduction?

The so-called "curse of dimensionality"

– Exponential growth in the number of examples required to accurately estimate a function

Exploratory data analysis

– Visualizing the structure of the data in a low-dimensional subspace

Two approaches to perform dimensionality reduction

- **Feature selection**: choose a subset of all the features

$$[x_1 \ x_2 ... x_N] \longrightarrow [x_{i_1} \ x_{i_2} ... x_{i_M}]$$

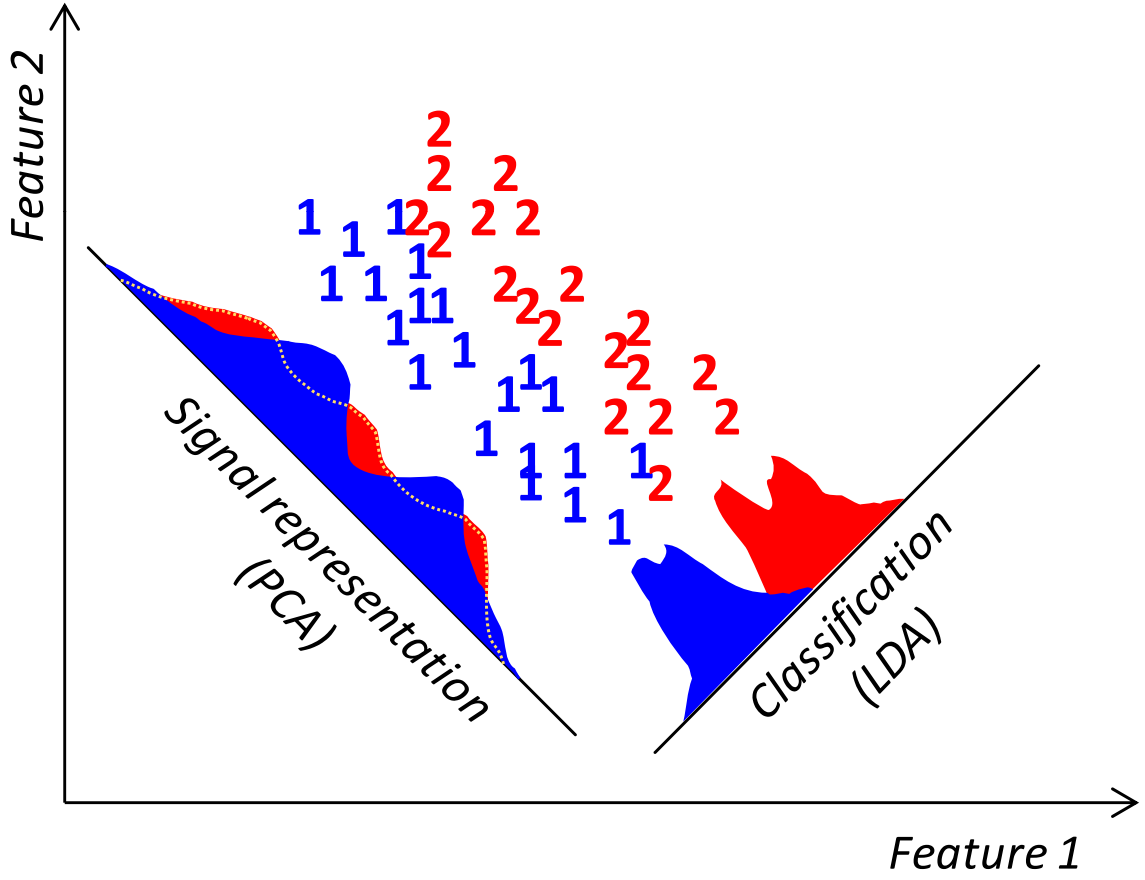
- **Feature extraction**: create new features by combining the existing ones

$$[x_1 \ x_2 ... x_N] \longrightarrow [y_1 \ y_2 ... y_M] = f([x_{i_1} \ x_{i_2} ... x_{i_M}])$$

- Feature extraction is typically a linear transform

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \xrightarrow{\text{linear feature extraction}} \begin{bmatrix} y_1 \\ y_2 \\ \\ y_M \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & & w_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \\ x_N \end{bmatrix}$$

Representation vs. classification



PCA

Solution

- Project the data onto the eigenvectors of the largest eigenvalues of the covariance matrix
- PCA finds orthogonal directions of largest variance

Properties

- If data is Gaussian, PCA finds independent axes
- Otherwise, it simply de-correlates the axes

Limitation

- Directions of high variance do not necessarily contain discriminatory information

LDA

Define scatter matrices

- Within class

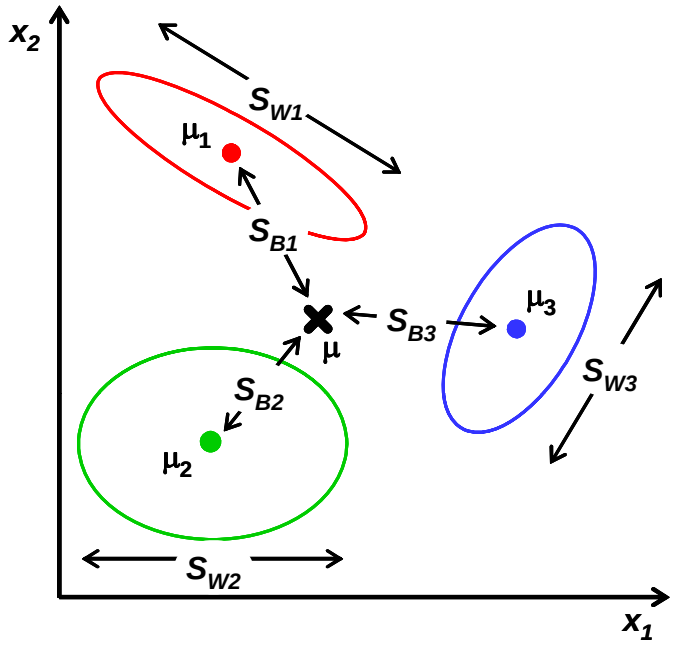
$$S_W = \sum_{i=1}^{C} S_i = \sum_{i=1}^{C} \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

- Between class

$$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

- Then maximize ratio

$$J(W) = \frac{\left| W^T S_B W \right|}{\left| W^T S_W W \right|}$$



Solution

- Optimal projections are the eigenvectors of the largest eigenvalues of the generalized eigenvalue problem

$$\left(S_B - \lambda_i S_W\right) w_i = 0$$

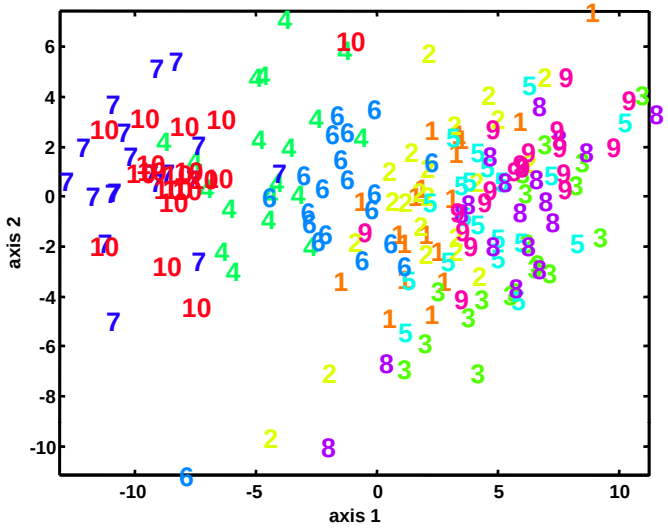
NOTE

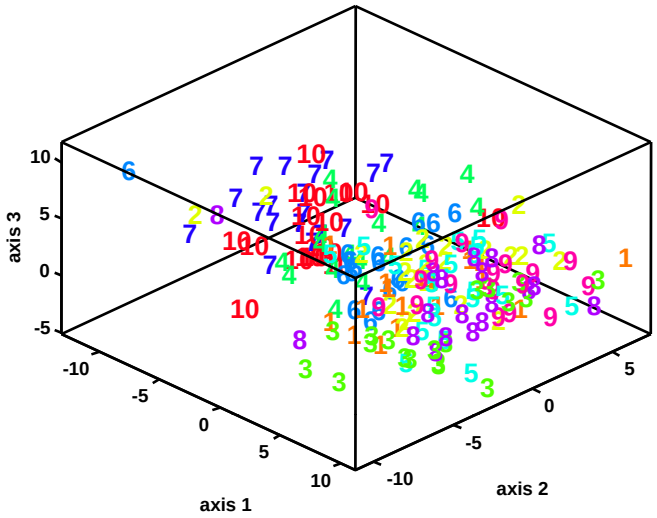
- S_B is the sum of C matrices of rank one or less and the mean vectors are constrained by $\Sigma\mu_i=\mu$
- Therefore, S_B will be at most of rank (C-1), and LDA produces at most C-1 feature projections

Limitations

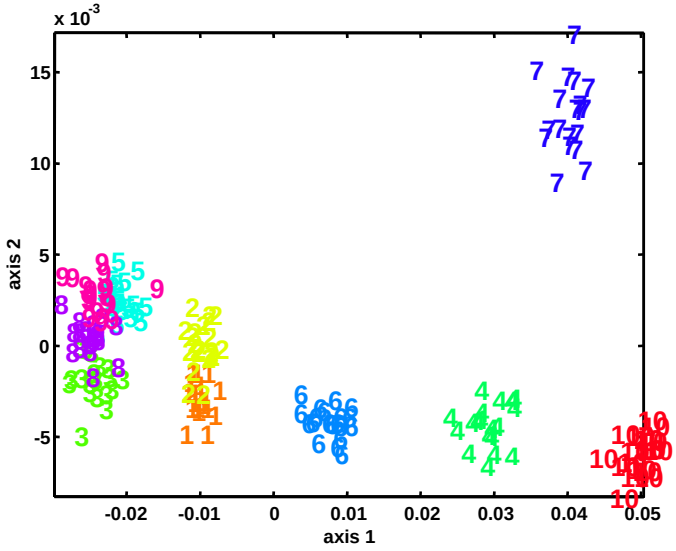
- Overfitting
- Information not in the mean of the data
- Classes significantly non Gaussian

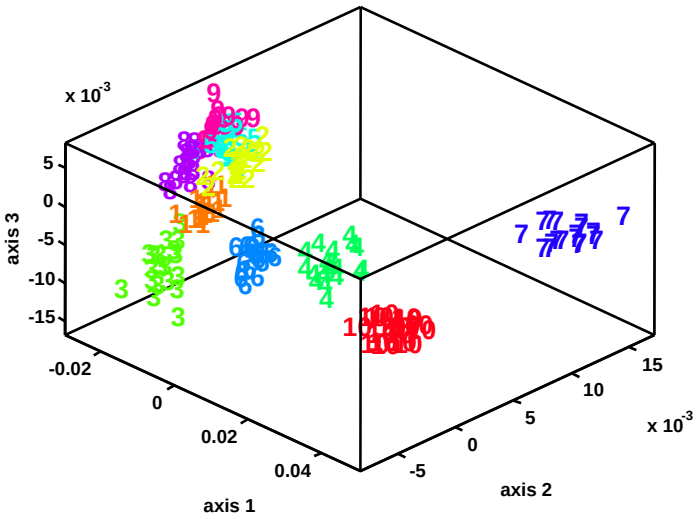
PCA


axis 1
axis 2


axis 1
axis 2
axis 3

LDA


x 10^-3
axis 1
axis 2


x 10^-3
axis 1
axis 2
axis 3
x 10^-3

LDA and overfitting

- Generate an artificial dataset

  Three classes, 50 examples per class, with the exact same likelihood: a multivariate Gaussian with zero mean and identity covariance