CHAPTER 3: BAYESIAN CLASSIFICATION

Grado en Ingeniería Informática

Curso 2014 / 15

Topics

1. Bayesian decision theory
    a. Bayesian classifier
2. Density estimation
3. Optimal decision boundary
4. Classifier of minimum prediction risk

THOMAS BAYES (1701-1761)

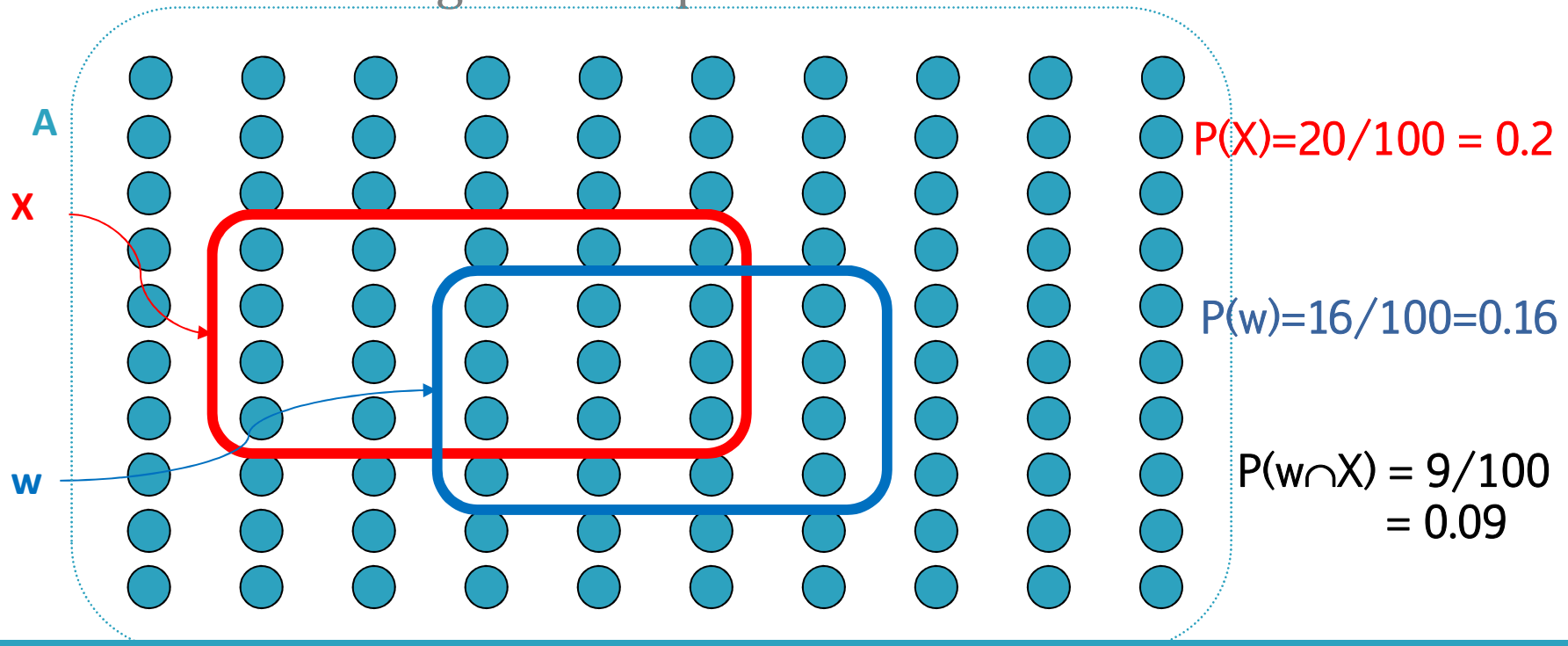
AN ESSAY TOWARDS SOLVING A PROBLEM IN THE DOCTRINE OF CHANCES.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY, 1763

BAYESIAN DECISION THEORY

Introduction

- Given *A*, a set of 100 elements with individual likelihood to be selected equal to 1/100
- *P(x)* is the likelihood of a element picked at random from the set *A* belongs to the space *X*

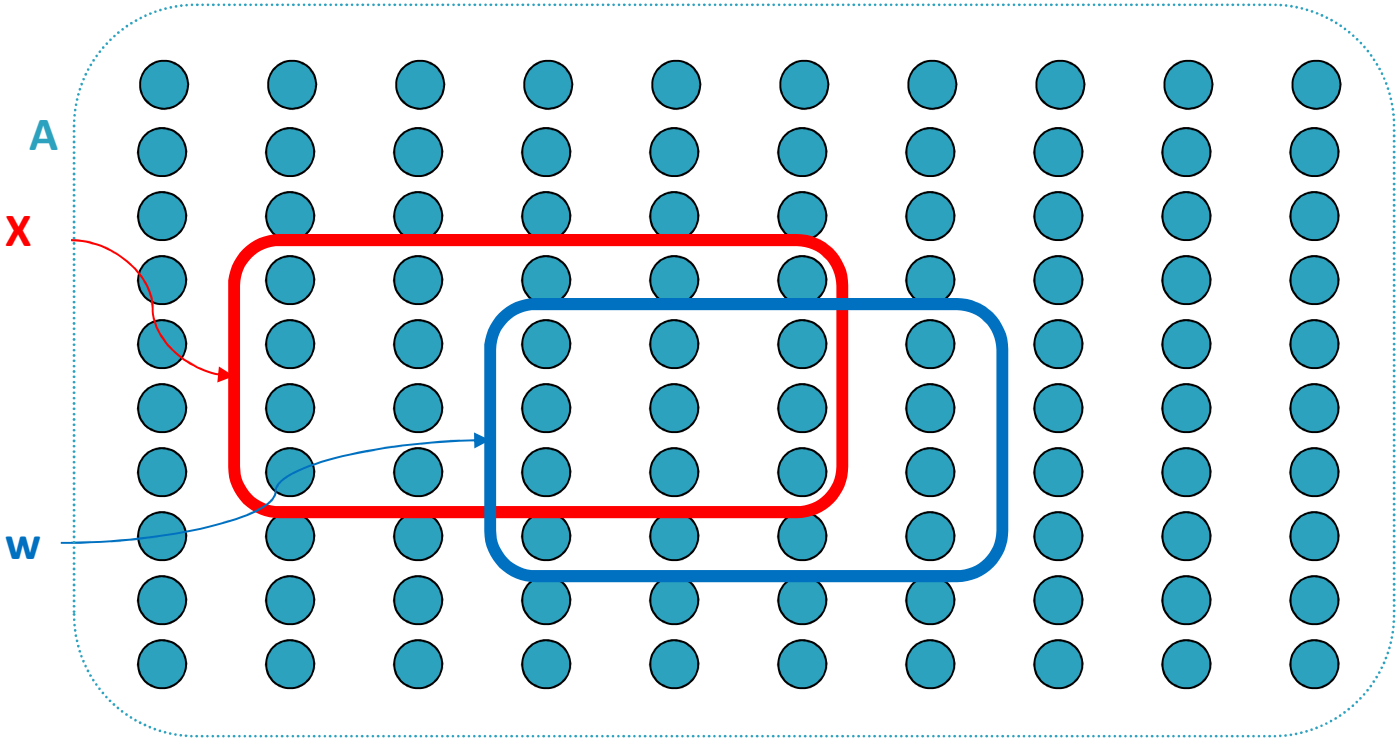


Introduction

- The probability of intersection is:

$$P(w \cap X) = P(w) \cdot P(X \mid w) = P(X) \cdot P(w \mid X)$$

- P(w∩X) = 0.16*(9/16) = 0.2 * (9/20) = 0.09



Bayes theorem

Conditional probability

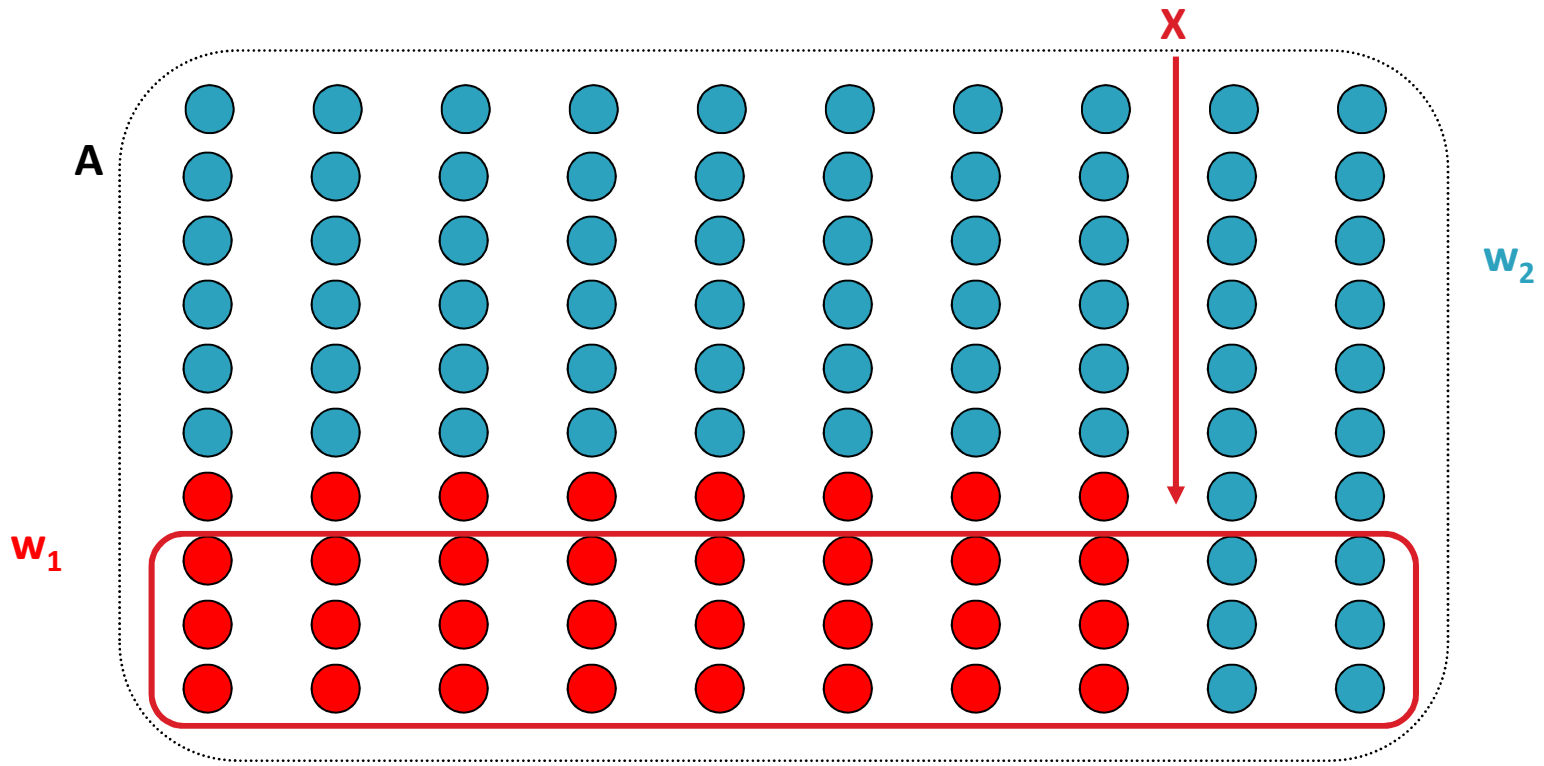
a priori probability

$$P(w \mid X) = \frac{P(X \mid w) \cdot P(w)}{P(X)}$$

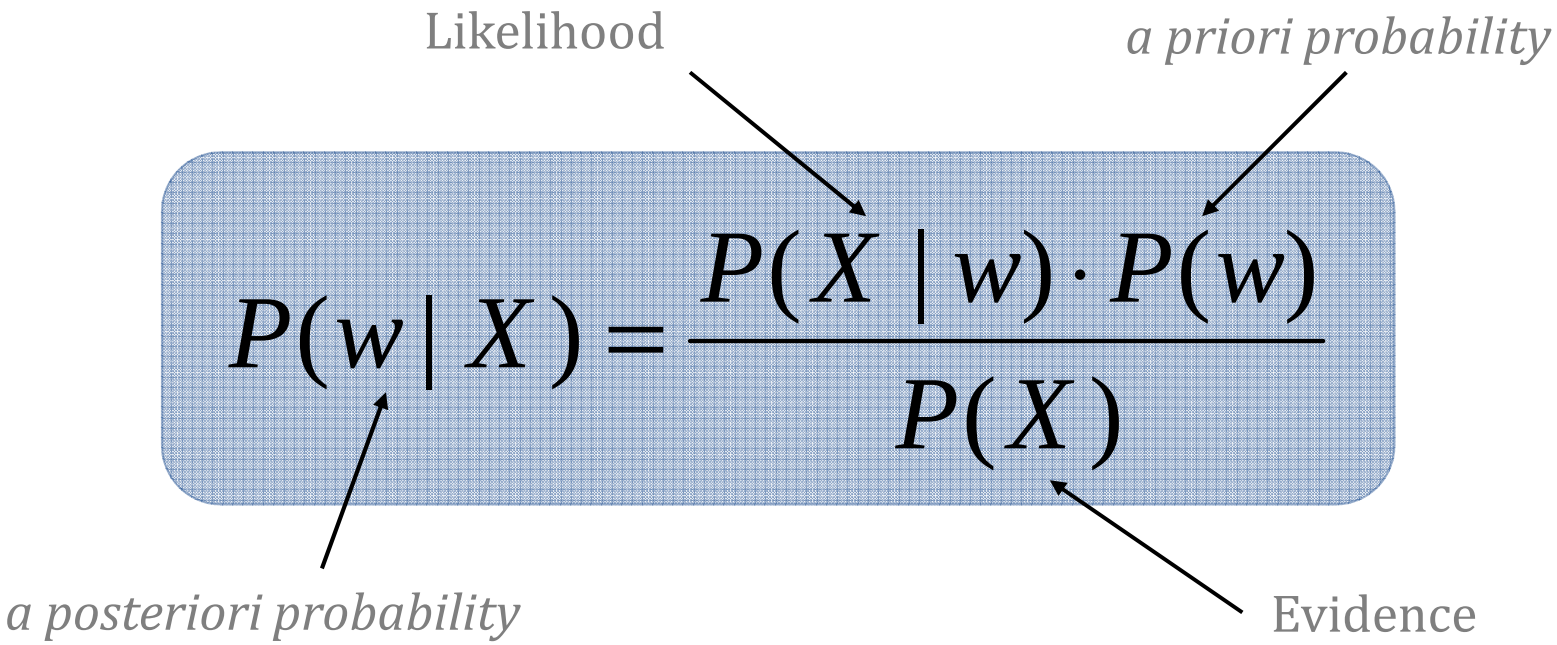
a posteriori probability

Bayes applied to classification

- A = charset studied
- X = a characteristic (for example: has earring)
- w1 = women set ● w2 = men set ●



Bayes theorem



- $P(X) = \sum_i P(X \mid w_i) \cdot P(w_i)$
- $\sum_i P(w_i \mid X) = 1$

A priori probability

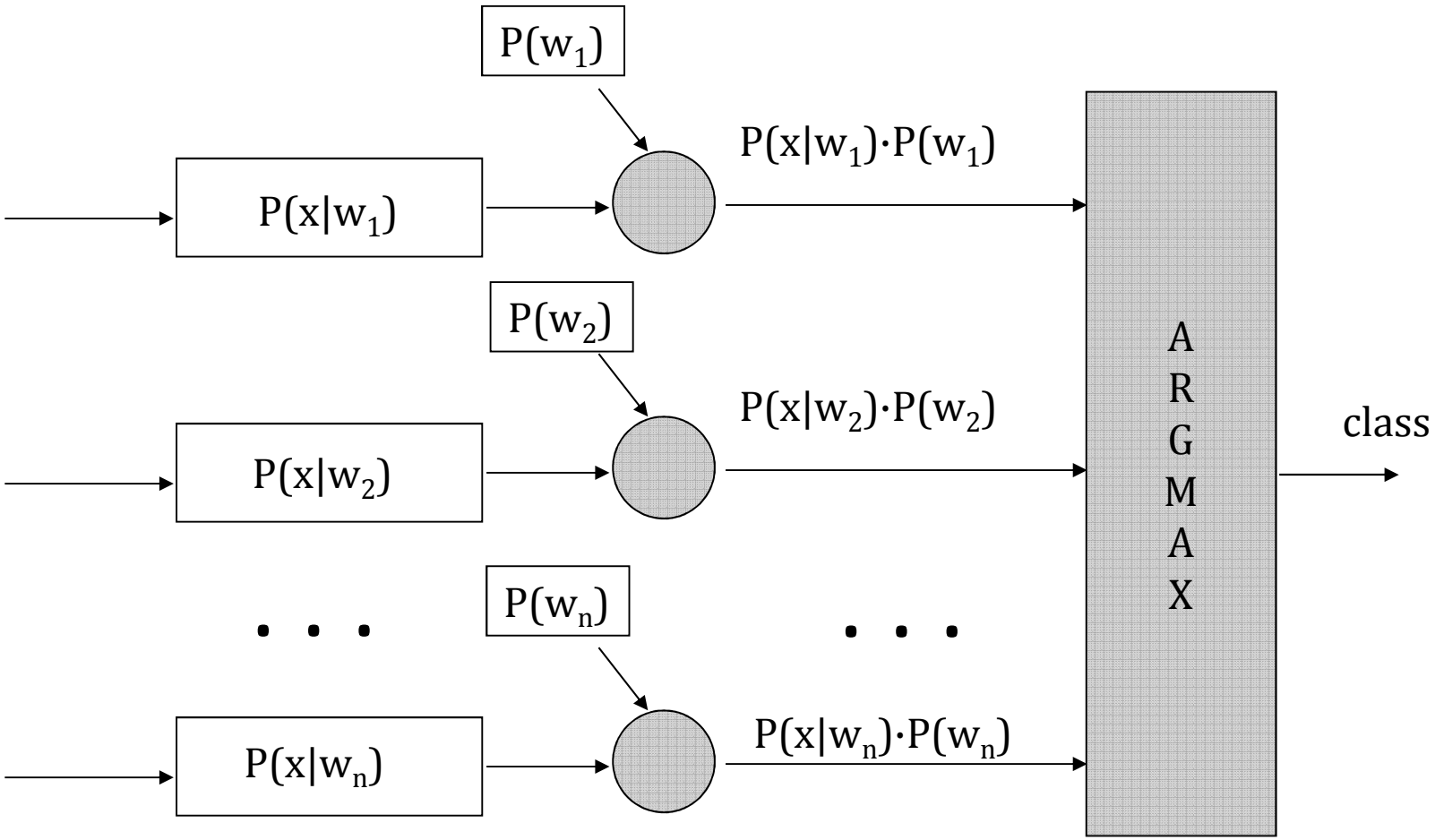
- Any person is selected

- Suppose I know the "a priori probability"

  P(man) = 68/100 P(woman) = 32/100

- If I have to decide about the sex knowing that this person has earring, I will say invariably that there would most likely be this person is a woman

Bayesian classifier



Bayesian classifier

- Suppose we know a data (X): has earrings

  P(w1) = P(man) = 68/100 P(w2) = P(woman) = 32/100

  P(X|w1) = P(has earrings | man) = 6/68

  P(X|w2) = P(has earrings | woman) = 24/32

  P(X) = P(has earrings) = P(X|w1)*P(w1)+P(X|w2)*P(w2) =

  = 6/68 * 68/100 + 24/32 * 32/100 = 30/100

- Applying the Bayes theorem:

  P(w1|X) = P(man | has earrings) = P(X|w1)*P(w1)/P(X)

  = (6/68*68/100)/(30/100) = 6/30 = 0.2

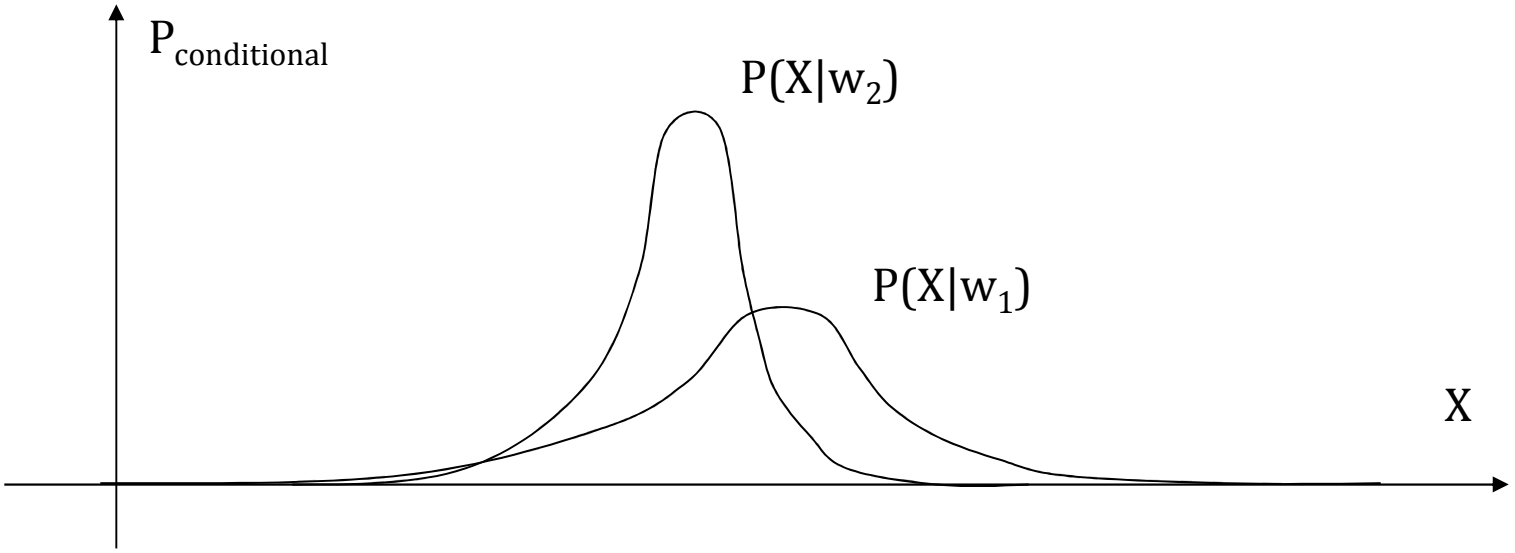
  P(w2|X) = P(woman | has earrings) = P(X|w2)*P(w2)/P(X)

  = (24/32*32/100)/(30/100) = 0.8

!!! So, there would most likely be this person is a woman!!!

Bayesian classifier

- *X* is often continuous value, therefore *P(X)* and $P(X|w_i)$ are probability density functions (for example: X = height)



How can we select a hypothesis?

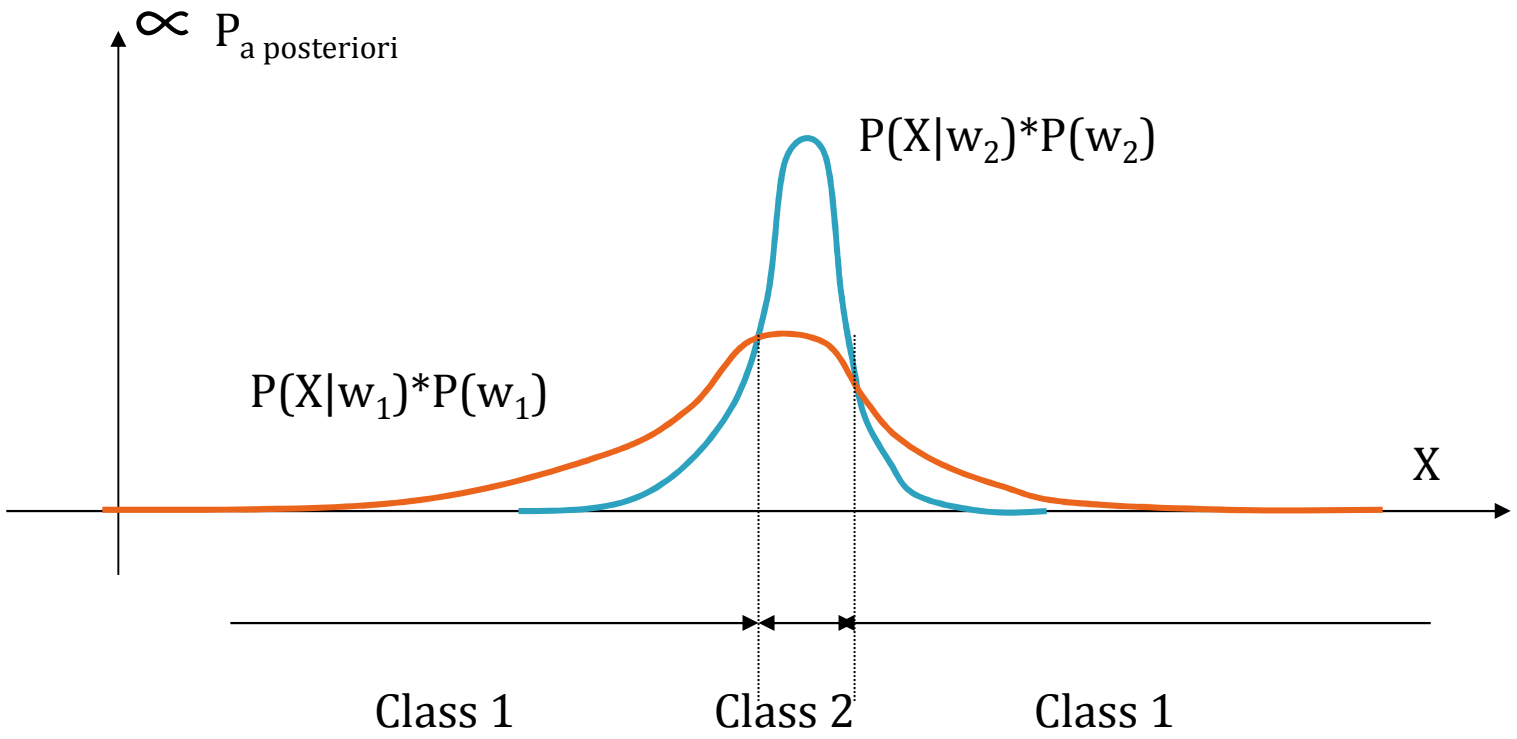
- MAP criteria (Maximum a - Posteriori)
  - Select the hypothesis with greater "a posteriori" probability
  - In the classification process, P(X) is equal in all hypothesis, so, we can remove it.

$$\text{argmax}(P(w_i|X)) = \text{argmax}(P(X|w_i)*P(w_i)/P(X)) = \text{argmax}(P(X|w_i)*P(w_i))$$

- Maximum likelihood criteria:
  - "A priori" equiprobable classes are considered

$$\text{argmax}(P(w_i|X)) = \text{argmax}(P(X|w_i))$$

How can we select a hypothesis?



DENSITY ESTIMATION

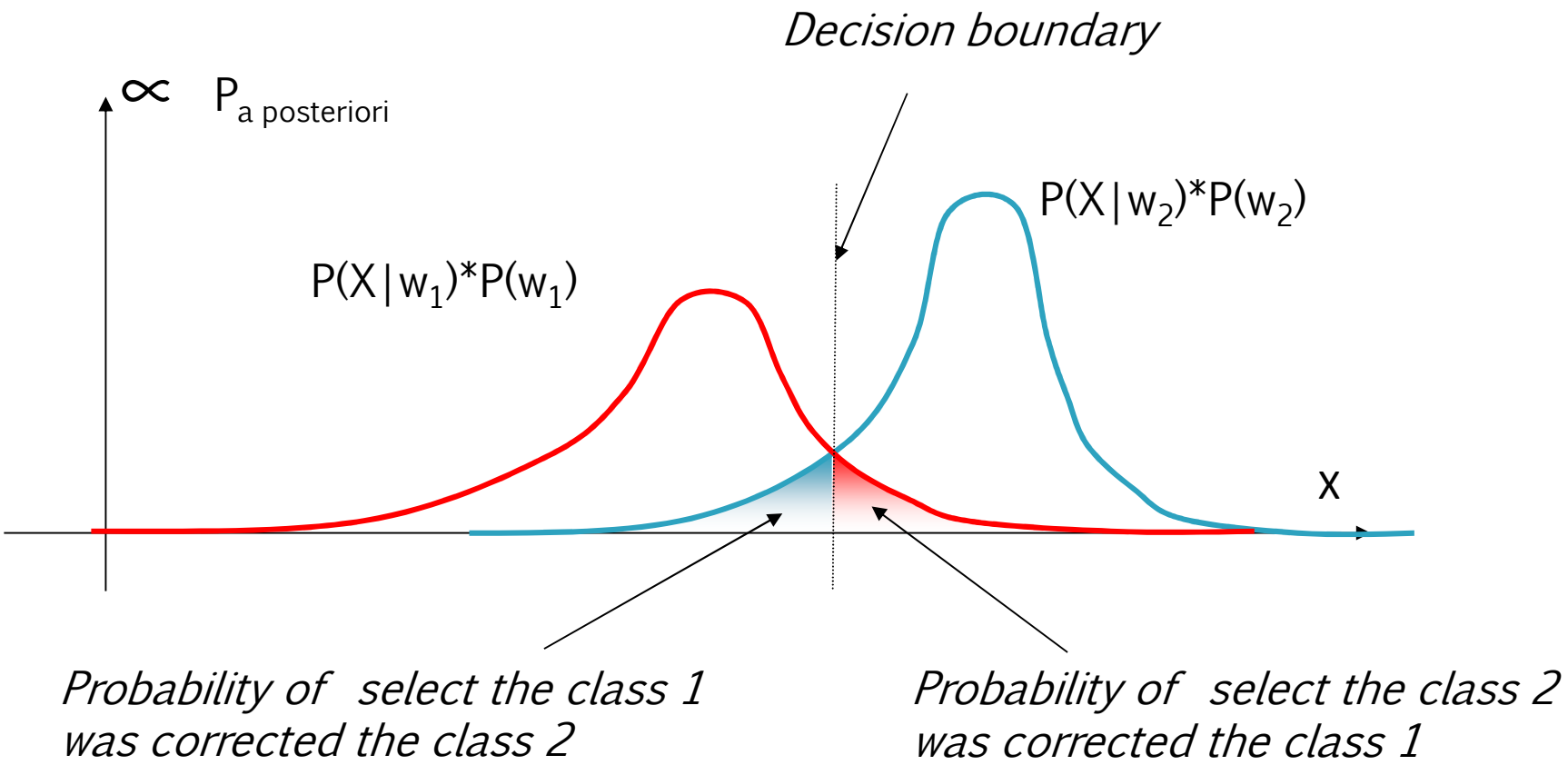
Density estimation

- To do a Bayes classifier is necessary to estimate, with high precision, these values:

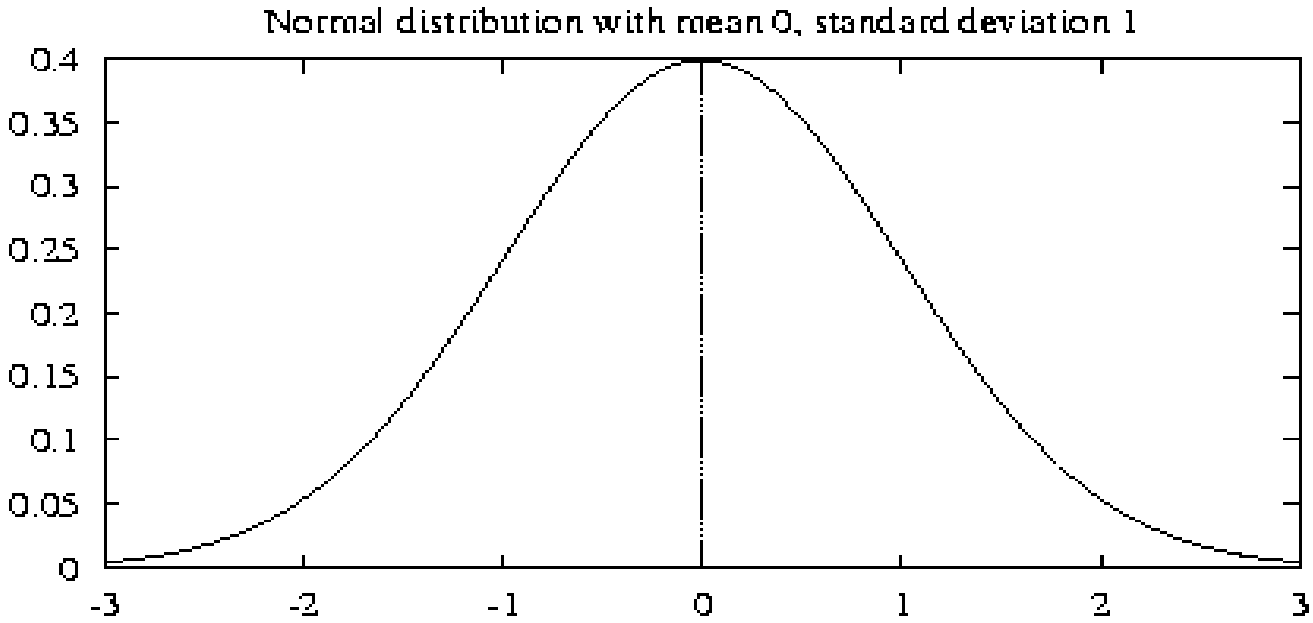
1. **$P(w_i)$:** they are often estimated simply by counting the data proportion in each class in the training data.

1. **$P(X|w_i)$:** It is a very complex area in pattern recognition, machine learning and statistical that we will discuss independently

Error in a hypothesis selection and determining the decision boundary (Other gives a higher error)



The normal distribution. Gaussian



Important parameters (Normal distribution)

- Density function

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Probability distribution

$$P(a \leq X < b) = \int_a^b p(x)\mathrm{d}x$$

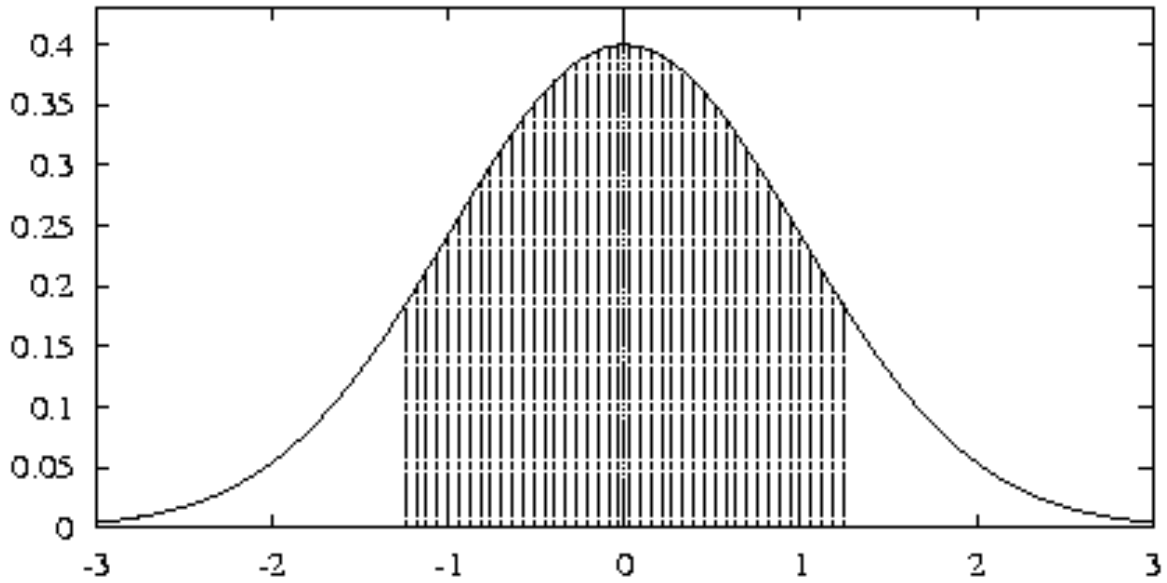
- Mean

$$E[X] = \mu$$

- Variance

$$Var[X] = \sigma^2$$

Percentages enclosed under the normal curve



N%	68.27%	95.45%	99.73%
Interval	$\mu \pm \sigma$	$\mu \pm 2 \cdot \sigma$	$\mu \pm 3 \cdot \sigma$

Density estimation

1. Parametric estimation:
   - It assumes that the distribution follows a known function (for example, Gaussian)
   - The estimation consists of determining the parameters of the function (for example, the mean and variance)

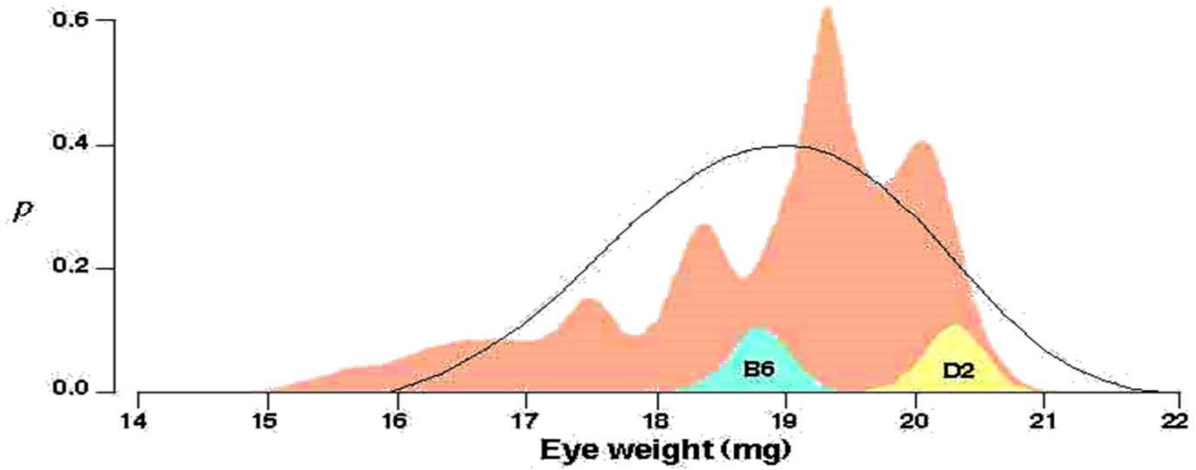
2. Non-parametric estimation
   - It makes no assumption about the underlying distribution in the data
   - An example may be the histogram. In this example, we have to determine the number of bars, position and width

Density estimation

- The usually functions are:
  - Rectangular
  - Gaussian
  - Exponential
  - Chi – square
  - Cauchy
  - …

Parametric estimation. Problems

- Data distribution rarely fits a simple mathematical function, such as a Gaussian.
- Parametric functions are unimodal (have a single maximum), and in practice, many problems are multi-modal.



Non – parametric estimation

- The probability that an item falls into a certain region R of the sample space is:

$$P = \int_{\Re} p(x') \cdot dx'$$

- If $p(x)$ is continuous and the R region is so small that $p(x)$ can be assumed constant in R, we can write:

$$\int_{\Re} p(x') \cdot dx' \cong p(x) \cdot V$$

where x is a point R, and V is the volume enclosed by R.

- If we have a sample of n data and k data falls into R, then the $k / (n \cdot V)$ fraction is a good estimate of the probability P

$$p(x) \cong \frac{k / n}{V}$$

Conditions for convergence

- The $k / (nV)$ fraction should be an average value of $p(x)$
- As V tends to 0, the fraction tends to $p(x)$
- In practice, V can not be too small, because as the number of data finite, k almost always be 0.

Conditions for convergence

- Theoretically, if the number of data is infinite, we can solve this problem

- To estimate the density of x, we form the sequence of regions R_1, R_2, ... containing x: the first region contains the first sample, the second, the second sample, etc.

- V_n is the volume of R_n, k_n and the number of samples that fall in R_n, and $p_n(x)$ the nth estimate of $p(x)$:

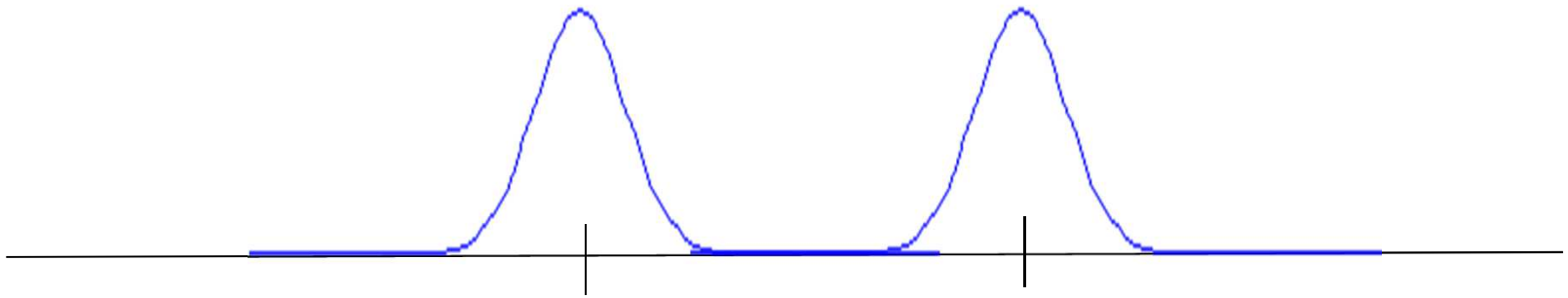
$$p_n(x) = (k_n/n)/V_n$$

Methods

- Histogram
  - Density obtained curve is not continuous
  - We have to set the width of the intervals
- Estimation of known density function (eg Gaussian)
  - The curve is continuous
  - Be estimated parameters (mean, variance) from function data
- Parzen window
  - The curve is continuous
  - The density curve is an analytic function, so it can not match a similar function to find the decision boundary analytically

Parzen method

1. Functions centered on each data (eg, Gaussian) are defined
2. This functions are added together
3. These are divided by the number of data used



Parzen method – Gaussian PDF

```
datos=shuffle([randn(1,1000)*5+30
               randn(1,1000)*3+17]);
desvstandard=0.5;
x=5:0.1:45;
h=zeros(1,length(x));
for i=1:length(datos),
  h = h + normpdf(x,datos(i), desvstandard);
  plot(x,h)
  title(num2str(datos(i)),'FontSize',14)
  drawnow
end
h = h / length(datos);
h2 = (normpdf(x,30,5) + normpdf(x,17,3))/2;

figure, plot(x,h,'r',x,h2,'b');
legend('calculada','exacta')
```

Parzen window estimator

- It assumes that the region R_n is a d-dimensional hypercube of side unit

$$V_n = h_n^d \ (\mathrm{h_n} : \text{side length } \Re_n)$$

$$\text{Given } \varphi(\mathrm{u}) \text{ the following function :}$$

$$\varphi(\mathrm{u}) = \begin{cases} 1 & \left|\mathrm{u_j}\right| \le \frac{1}{2} \quad \mathrm{j} = 1,\ldots,\mathrm{d} \\ 0 & \text{in other case} \end{cases}$$

- $\varphi((x-x_i)/h_n)$ is equal to 1 if x_i falls within the hypercube of volume equal to V_n centered at x, and 0 otherwise.

Parzen window estimator

- The samples number in the hypercube is:

$$k_n = \sum_{i=1}^{i=n} \varphi\left(\frac{x - x_i}{h_n}\right)$$

- Replacing k_n, this expression is obtained:

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{i=n} \frac{1}{V_n} \varphi\left(\frac{x - x_i}{h_n}\right)$$

  - The functions can be φ other !!

    $\varphi(\boldsymbol{u}) \sim N(0, \boldsymbol{I})$

  - This avoids discontinuities

Parzen window estimator

- Example:
  - $p(x) \rightarrow N(0,1)$

    Given $\varphi(u) = (1/\sqrt{(2\pi)}) \exp(-u^2/2)$ and $h_n = h_1/\sqrt{n}$ $(n>1)$

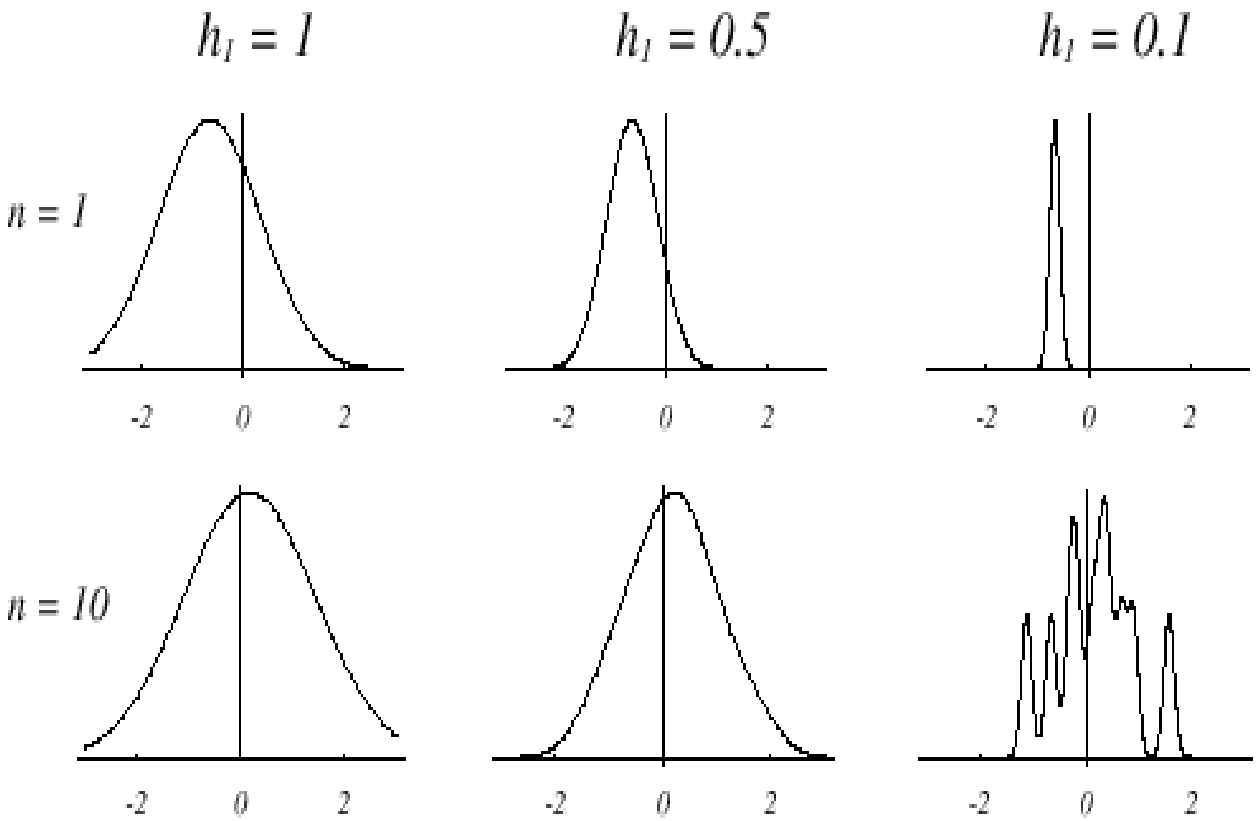
    (h_1: known parameter)

    Then:

$$p_n(x) = \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h_n}\varphi\left(\frac{x - x_i}{h_n}\right)$$

    is an average of densities normal centered samples x_i.

Parzen window estimator



Parzen window estimator

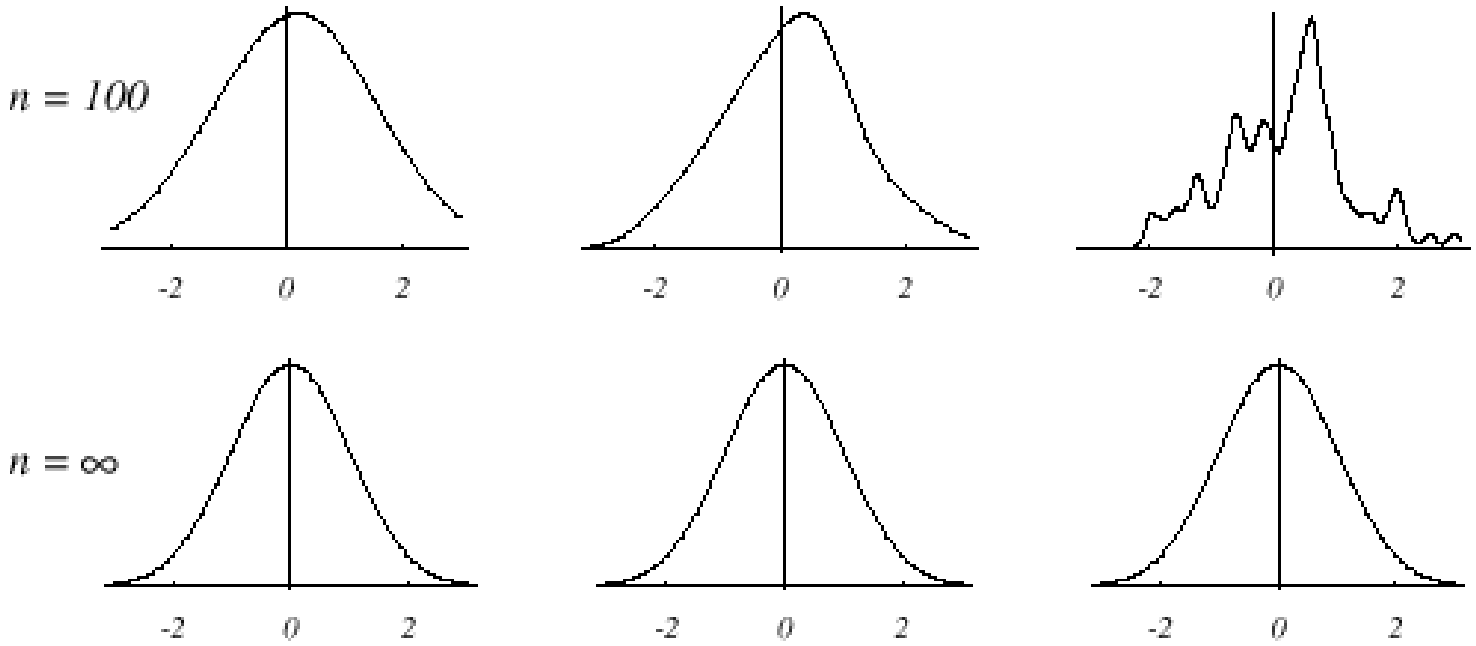
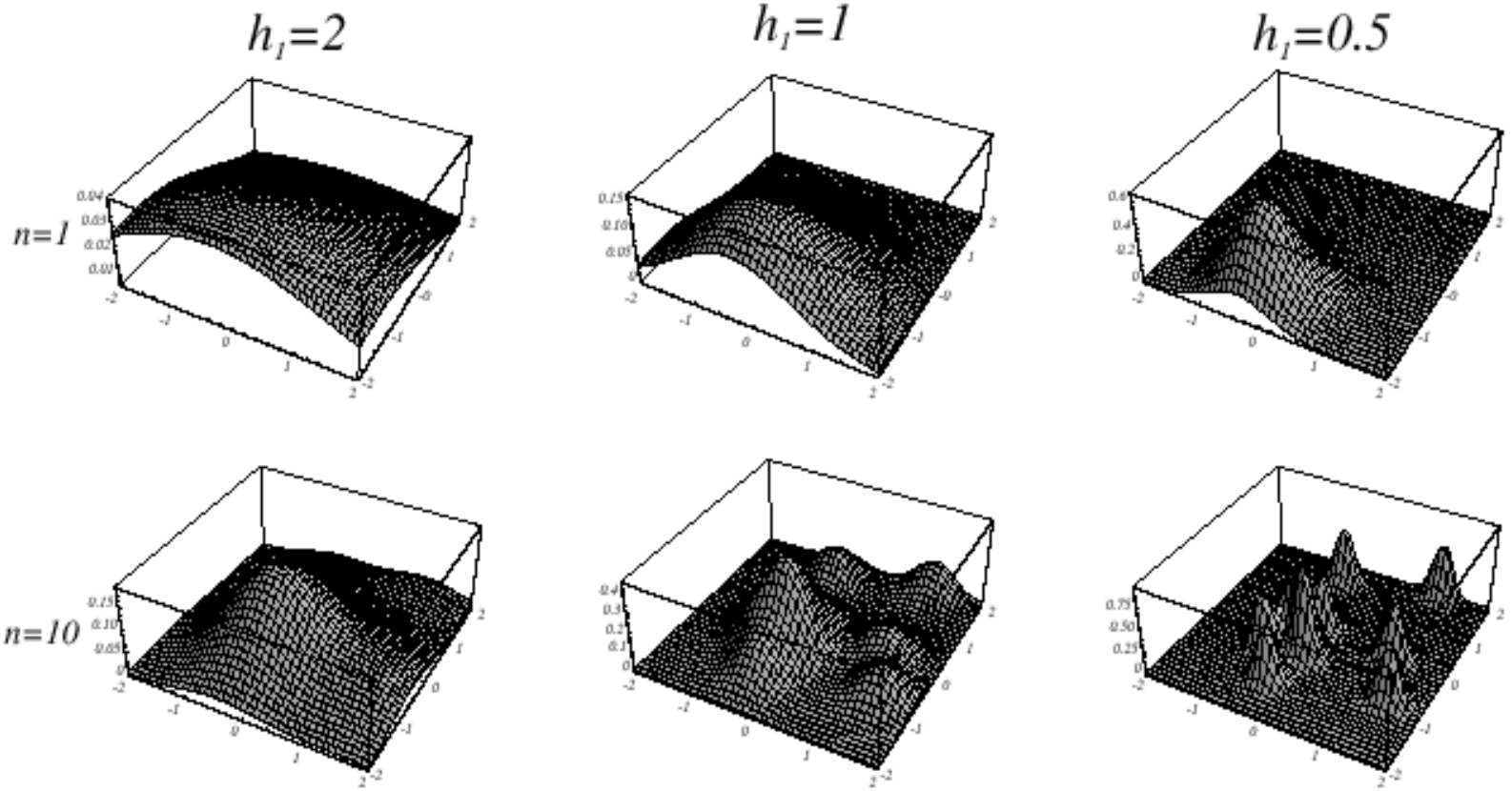


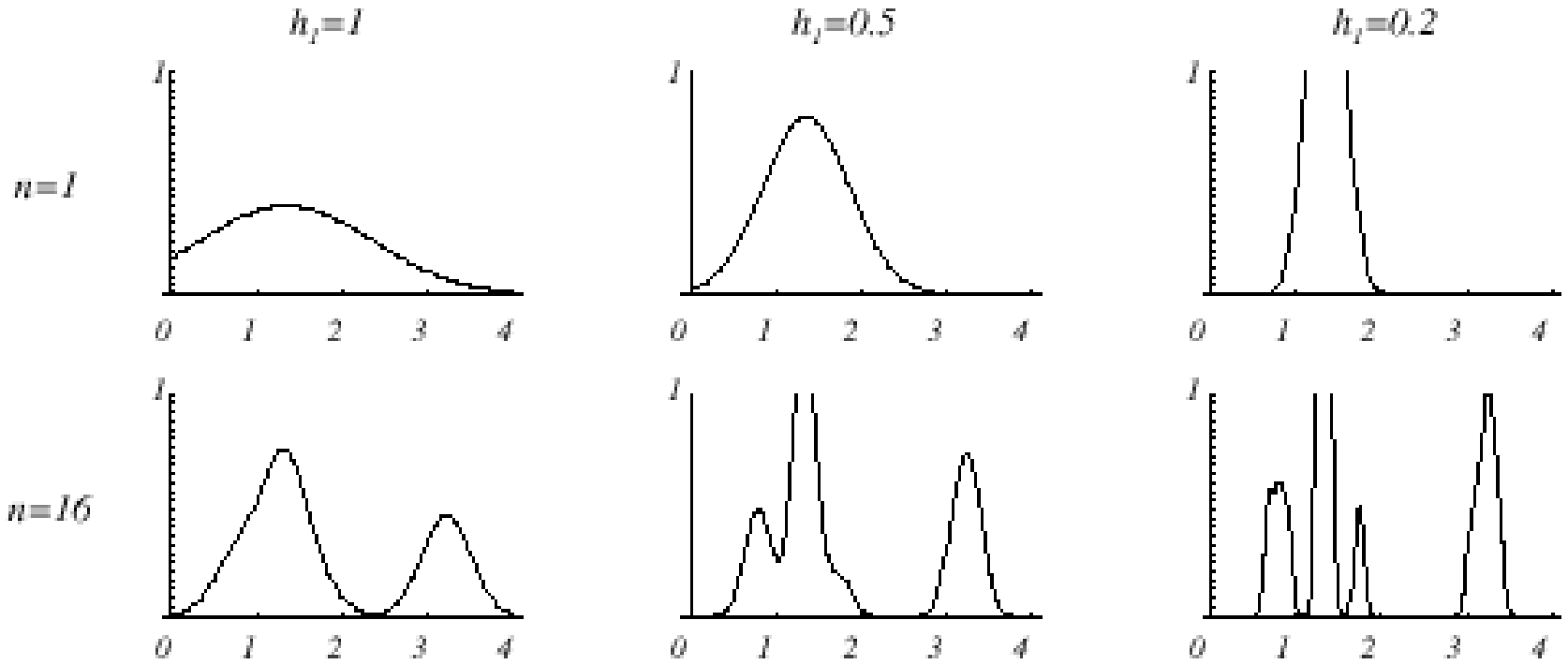
FIGURE 4.5. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Parzen window estimator

- Two dimensions may also be applied:



Parzen window estimator



Parzen window estimator

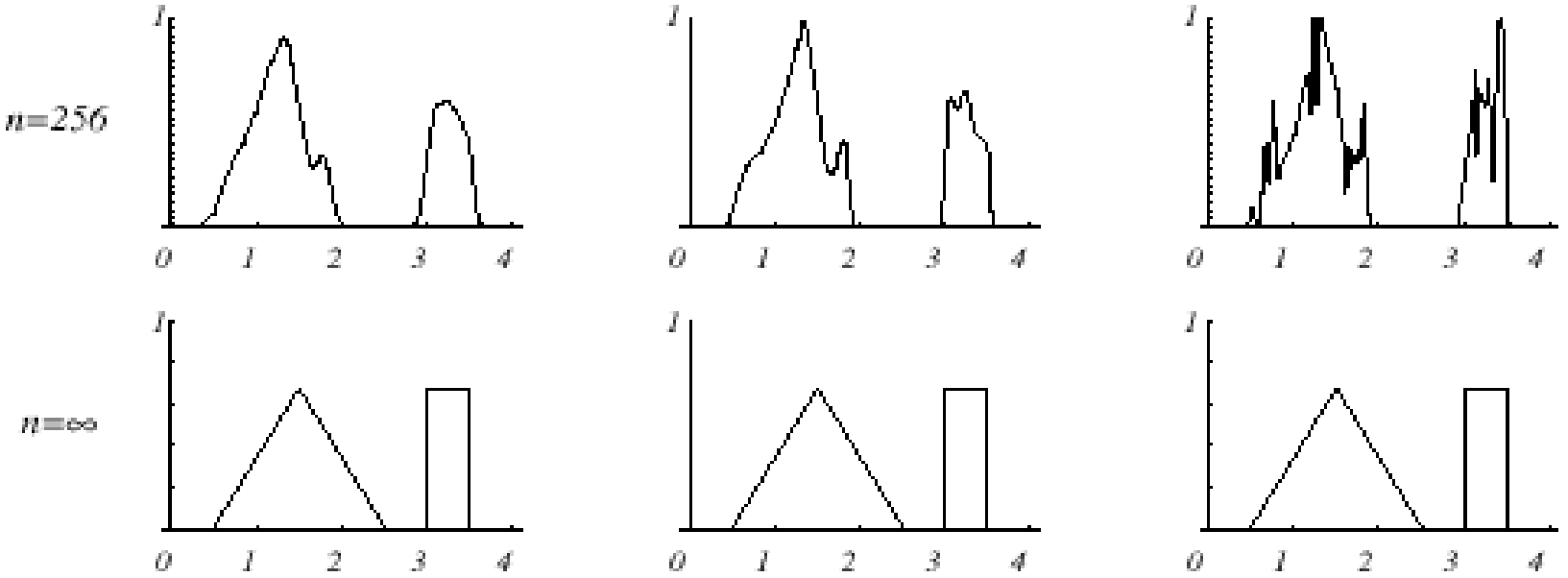


FIGURE 4.7. Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Parzen window estimator

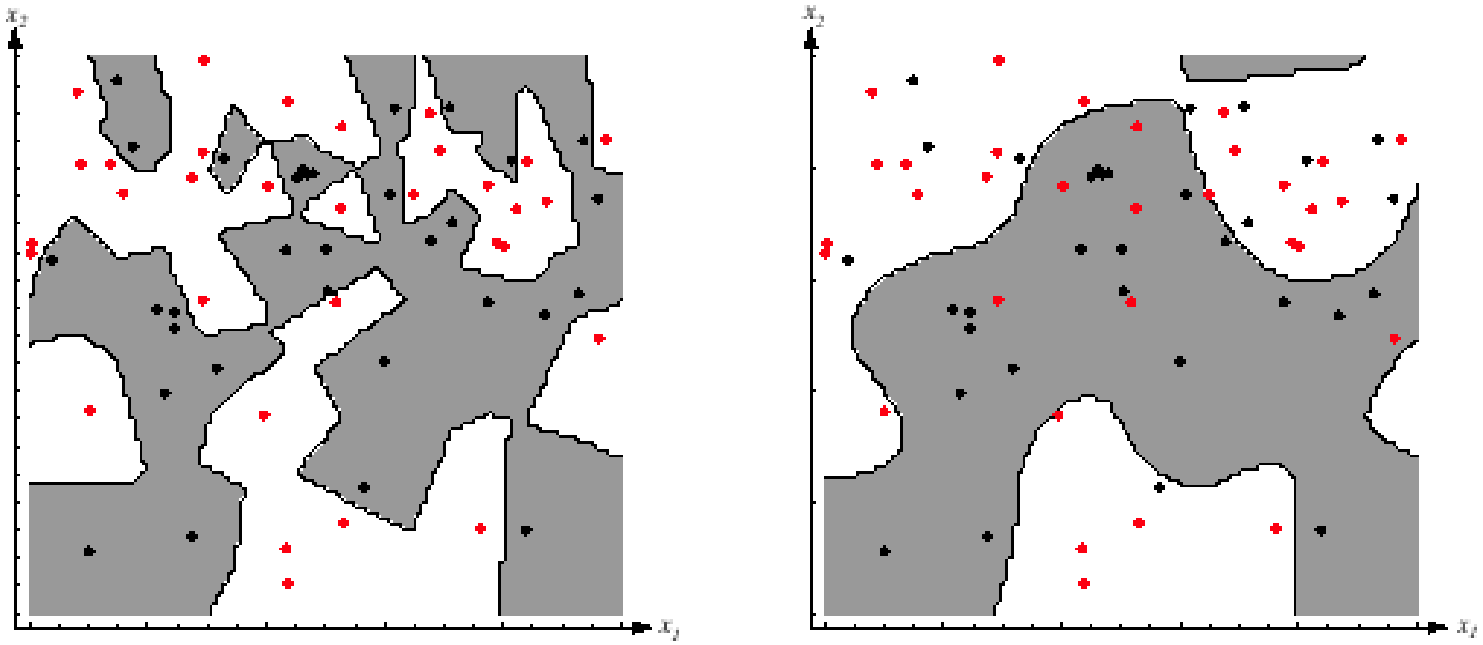


FIGURE 4.8. The decision boundaries in a two-dimensional Parzen-window dichotomizer depend on the window width h. At the left a small h leads to boundaries that are more complicated than for large h on same data set, shown at the right. Apparently, for these data a small h would be appropriate for the upper region, while a large h would be appropriate for the lower region; no single window width is ideal overall. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Semi-parametric methods

- They consist of somewhere between parametric and nonparametric methods

- They are based on parametric methods applied at various points centered calculated nonparametrically space from the data

- To apply these methods must be known a clustering algorithm, for example, the k-means.

K-Means method (clustering)

- K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem.

- The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori.

- This algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

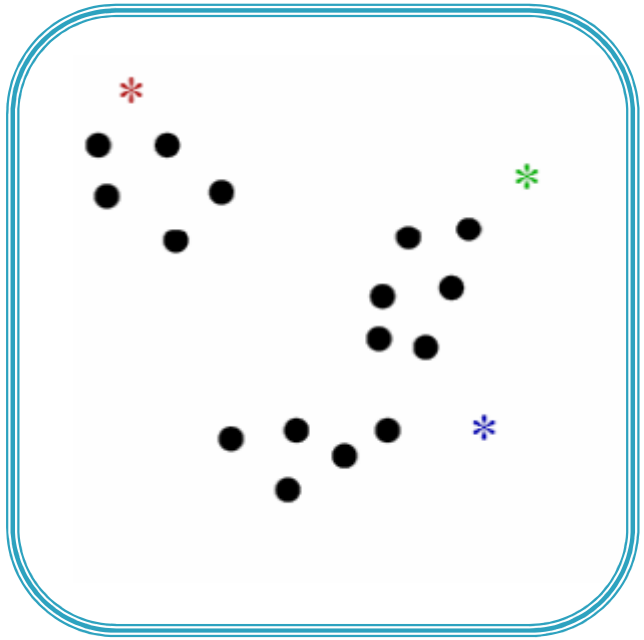
where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point and the cluster center , is an indicator of the distance of the n data points from their respective cluster centers.

K-Means method (clustering)

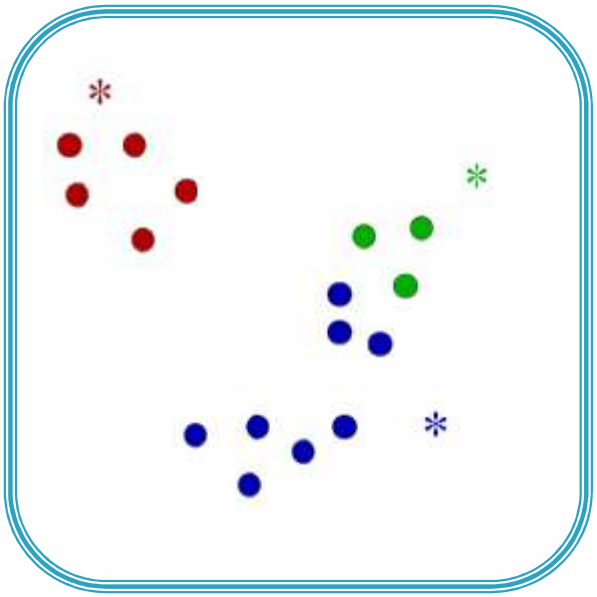
1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids
2. Assign each object to the group that has the closest centroid
3. When all objects have been assigned, recalculate the positions of the K centroids
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

K-Means method (clustering)

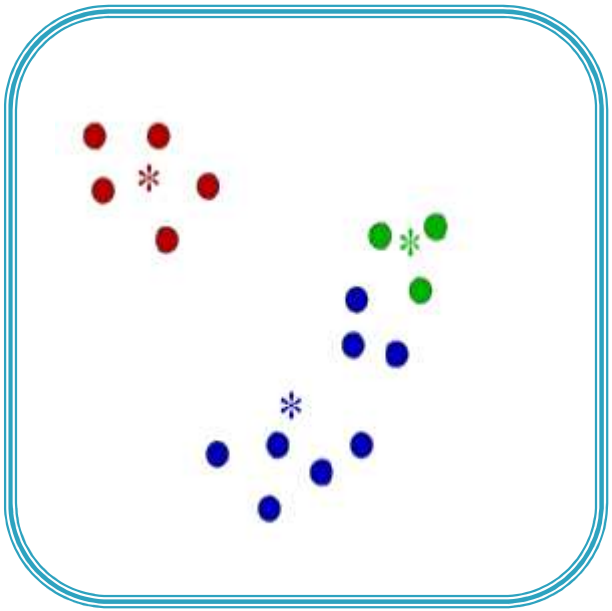
- K = # of clusters (given); one “mean” per cluster
- Initialize means (e.g. by picking k samples at random)
- Iterate:
  - (1) assign each point to nearest mean
  - (2) move “mean” to center of its cluster.



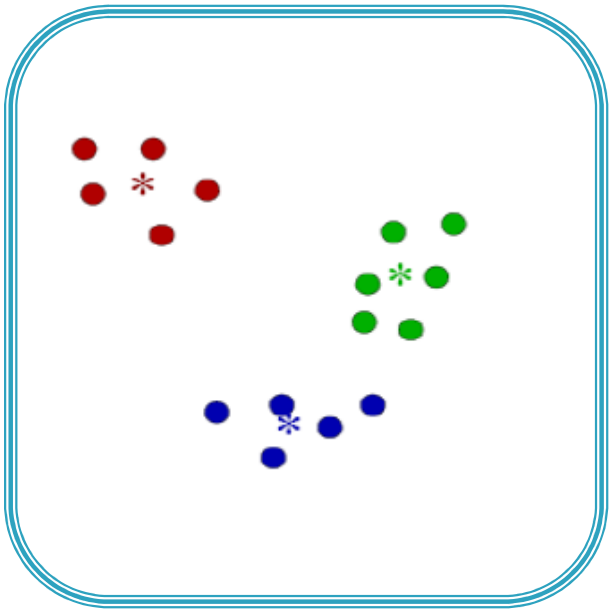
K-Means method (clustering)



Assign to nearest representative



Re – estimate means



Convergence

OPTIMAL DECISION BOUNDARY

Introduction

1. Defining p.d.f equation of Class 1

2. Defining p.d.f equation of Class 2

3. Equalizing the two equations and solve the equation in the variable x.

a) Two equiprobable classes. Normal distribution 1D , $\sigma_1=\sigma_2=\sigma$

- We suppose that:
  a) $P(X|w_1)=N(\mu_1,\sigma^2)$ $P(X|w_2)=N(\mu_2,\sigma^2)$
  b) $P(w_1) = P(w_2) = 0.5$
- The decision boundary must satisfy that:

$$\mathbf{P(X|w_1)*P(w_1)=P(X|w_2)*P(w_2)}$$

- Applying logarithms:

$$\ln(P(w_1)) - 0.5\cdot\ln(2\pi\sigma^2) - \frac{(x-\mu_1)^2}{2\sigma^2} = \ln(P(w_2)) - 0.5\cdot\ln(2\pi\sigma^2) - \frac{(x-\mu_2)^2}{2\sigma^2}$$

- And developing the equation:

$$(x-\mu_1)^2 = (x-\mu_2)^2$$

b) Two classes, Normal distribution 1D

- The decision boundary must satisfy that:

$$\mathbf{P(X|w_1)*P(w_1)=P(X|w_2)*P(w_2)}$$

- Applying logarithm and removing the same terms:

$$\ln(P(w_1)) - \ln(\sigma_1) - \frac{(x-\mu_1)^2}{2\sigma_1^2} = \ln(P(w_2)) - \ln(\sigma_2) - \frac{(x-\mu_2)^2}{2\sigma_2^2}$$

- Developing, the solution is the equation $Ax^2+Bx+C=0$, where:

$$A = \sigma_1^2 - \sigma_2^2$$
$$B = 2\left(\mu_1 \cdot \sigma_2^2 - \mu_2 \cdot \sigma_1^2\right)$$
$$C = \left[\ln(P(w_1)) - \ln(\sigma_1) - \ln(P(w_2)) + \ln(\sigma_2)\right] \cdot 2 \cdot \sigma_1^2 \cdot \sigma_2^2 + \left(\sigma_1^2 \cdot \mu_2^2 - \sigma_2^2 \cdot \mu_1^2\right)$$

Matlab example

```
m1 = 3; s1= 2;  m2 = 5; s2=3;
x=[m1+s1*randn(1,30) m2+s2*randn(1,20)];
y=[ones(1,30) 2*ones(1,20)];

indices1=find(y==1);  indices2=find(y==2);
m1=mean(x(indices1)); m2=mean(x(indices2));
s1=std(x(indices1));  s2=std(x(indices2));
Pw1=length(indices1)/length(y);
Pw2=length(indices2)/length(y);

A=s1*s1-s2*s2;
B=2*(m1*s2*s2-m2*s1*s1);
C=2*s1*s1*s2*s2*(log(Pw1)-log(Pw2)-
  log(s1)+log(s2))+s1*s1*m2*m2-s2*s2*m1*m1;
x1=(-B+sqrt(B*B-4*A*C))/2/A
x2=(-B-sqrt(B*B-4*A*C))/2/A

I=-9:0.01:9;plot(I,Pw1*normpdf(I,m1,s1));hold on;
plot(I,Pw2*normpdf(I,m2,s2),'r');hold off;
```

Multivariate normal distribution

- Density function:

$$f_{\mathbf{x}}(x) = \frac{1}{(2\pi)^{n/2} \cdot |\Sigma|^{1/2}} \cdot e^{-\frac{1}{2} \cdot (x-\boldsymbol{\mu})^T \cdot \Sigma^{-1} \cdot (x-\boldsymbol{\mu})}$$

- Σ_{nxn} = Covariance matrix, defined as:

$$\Sigma = \mathrm{E}\left[(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})^T\right] = \begin{bmatrix} \sigma_1^2 & c_{12} & \cdots & c_{1n} \\ c_{12} & \sigma_2^2 & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{1n} & c_{2n} & \cdots & \sigma_n^2 \end{bmatrix}$$

$$c_{ij} = \mathrm{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right]$$

Multivariate normal distribution

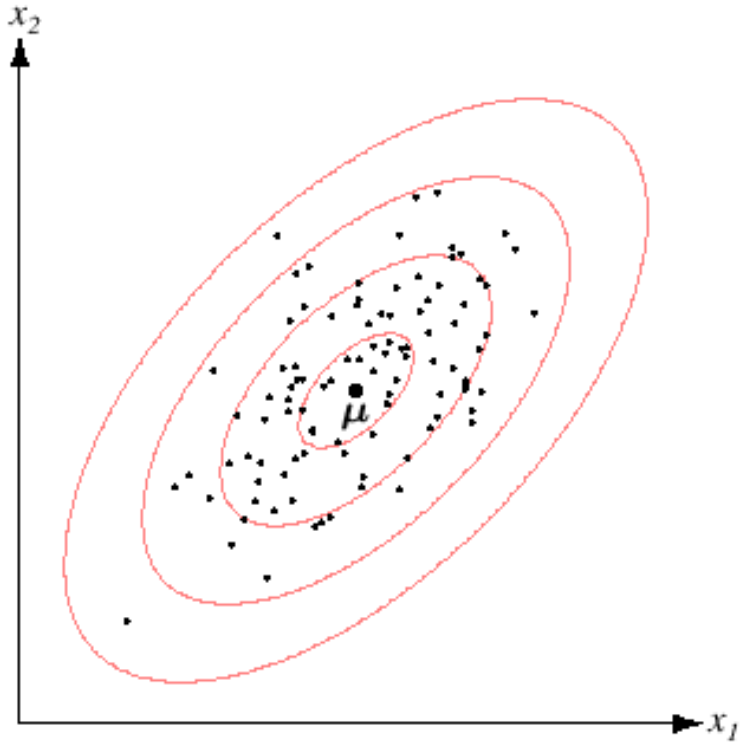
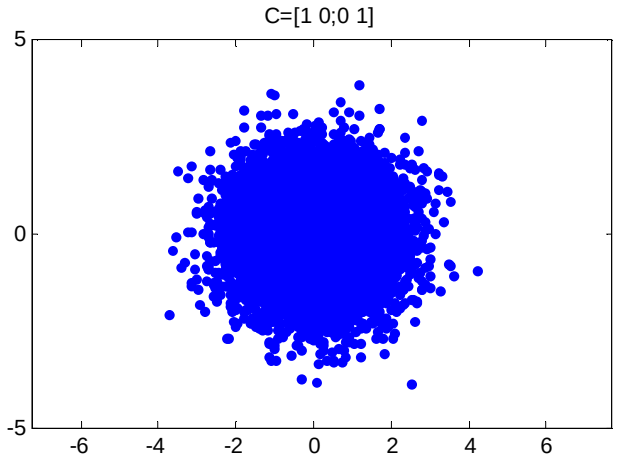
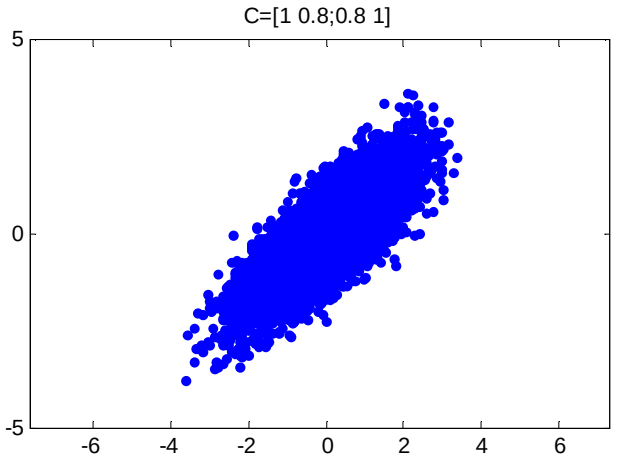
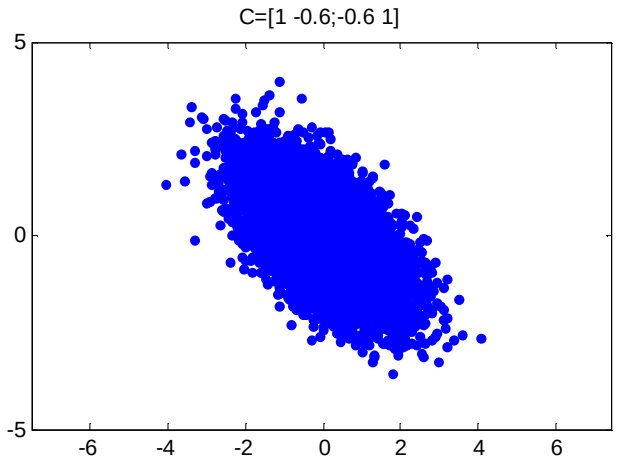


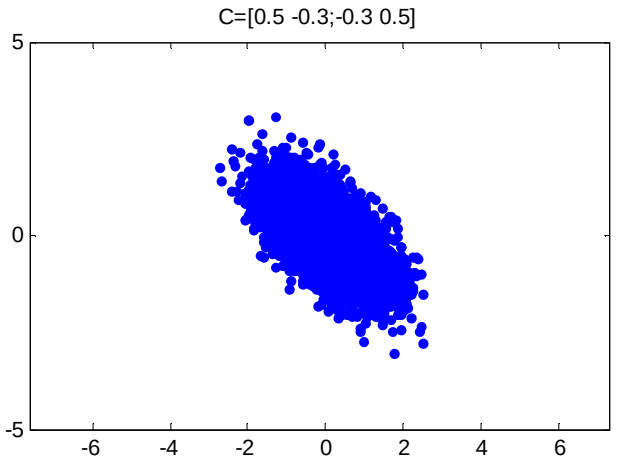
FIGURE 2.9. Samples drawn from a two-dimensional Gaussian lie in a cloud centered on the mean $\boldsymbol{\mu}$. The ellipses show lines of equal probability density of the Gaussian. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Covariance matrix examples

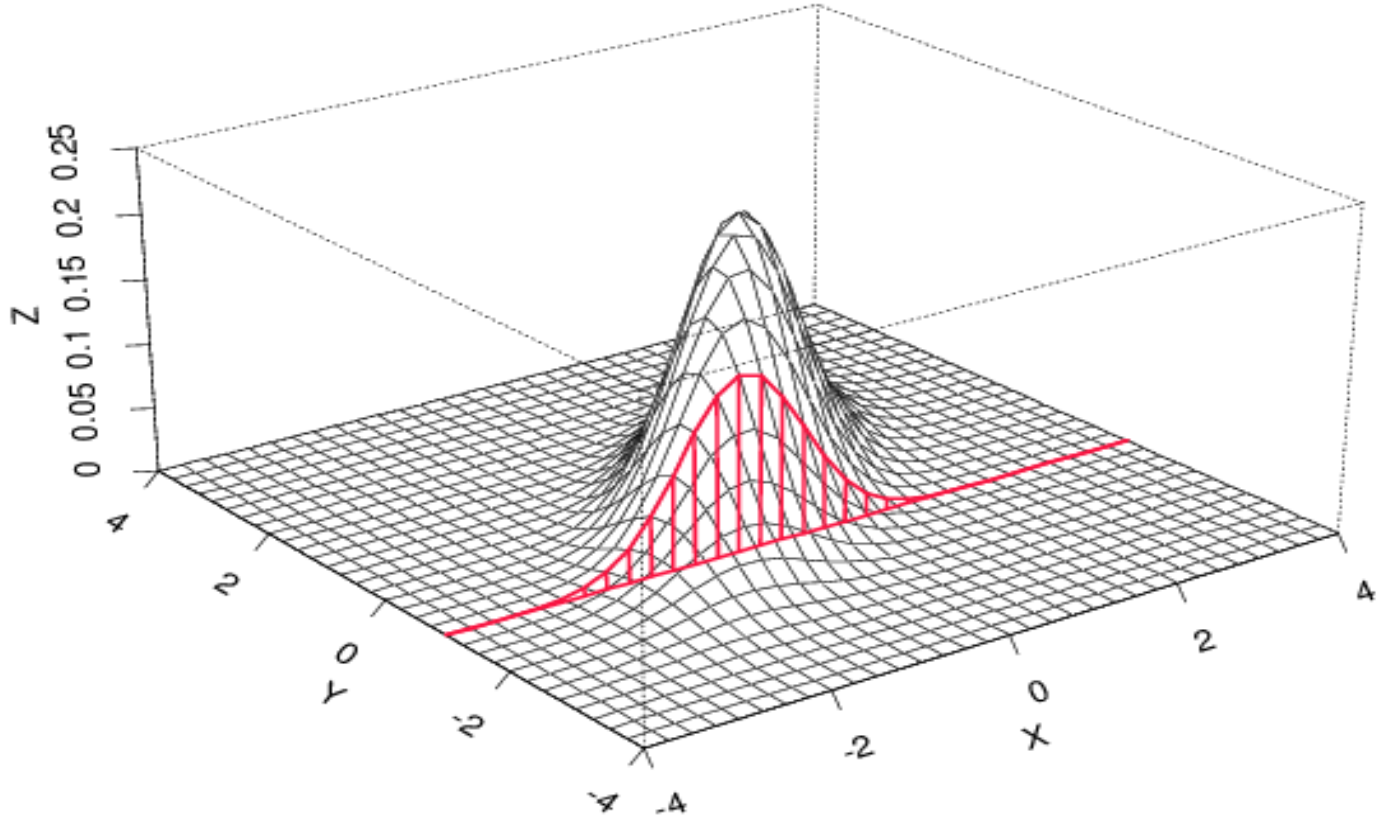








Multivariate normal distribution (in this case bivariate)



In red, P(X|Y=-1) is drawn

How to generate data and visualize 2D Gaussian pdf?

```
[x,y]=meshgrid(-3:0.1:3);
m1 = [0;0];      C1 = [1 -0.9;-0.9 1];
m2 = [0.5;0.9]; C2 = [0.7 0.3;0.3 0.7];

D1 = mvnpdf([x(:) y(:)],m1',C1);
D1=reshape(D1,size(x));
D2 = mvnpdf([x(:) y(:)],m2',C2);
D2=reshape(D2,size(x));

close all
mesh(D1);hold on;
mesh(D2);hold off;
figure, mesh(double(D1>D2))
```

Use either of these two functions:

- **mvnpdf**
- **randnorm**

2D Gaussians – Covariance matrix

- It is the generalization to higher dimensions of the concept of variance
- The covariance matrix is a matrix which contains the variance between the elements of a vector

$$\Sigma = \begin{bmatrix} \mathrm{E}[(X_1 - \mu_1)(X_1 - \mu_1)] & \mathrm{E}[(X_1 - \mu_1)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_1 - \mu_1)(X_n - \mu_n)] \\ \mathrm{E}[(X_2 - \mu_2)(X_1 - \mu_1)] & \mathrm{E}[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_2 - \mu_2)(X_n - \mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n - \mu_n)(X_1 - \mu_1)] & \mathrm{E}[(X_n - \mu_n)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_n - \mu_n)(X_n - \mu_n)] \end{bmatrix}.$$

2D Gaussians

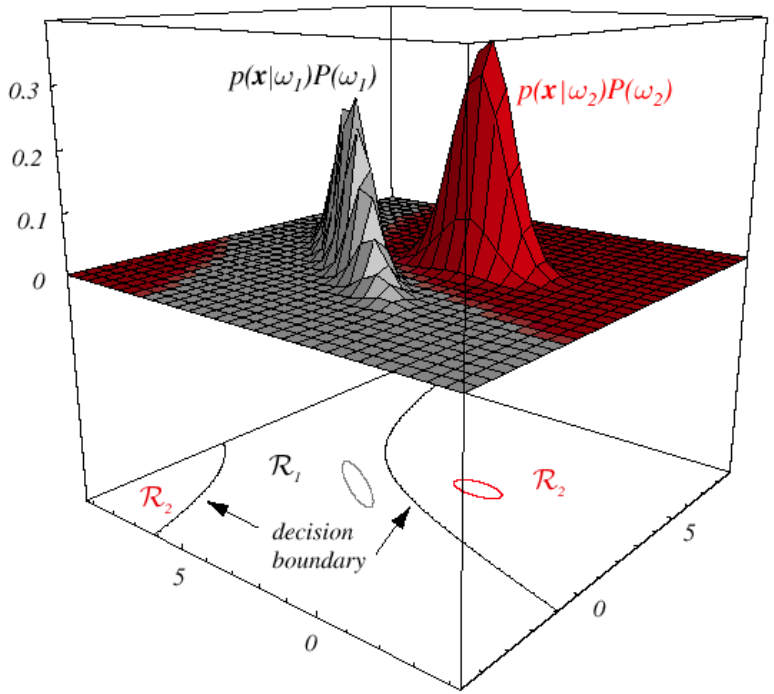


FIGURE 2.6. In this two-dimensional two-category classifier, the probability densities are Gaussian, the decision boundary consists of two hyperbolas, and thus the decision region $\mathcal{R}_2$ is not simply connected. The ellipses mark where the density is $1/e$ times that at the peak of the distribution. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Optimal boundary 2 equiprobable classes, N-dimensional normal distribution, C1 = C2 = C

- We supose that:
  a) $P(X|w_1)=N(\mu_1,C)$; $P(X|w_2)=N(\mu_2,C)$
  b) $P(w_1) = P(w_2) = 0.5$
- The decision boundary must satisfy that:

$$\mathbf{P(X|w_1)*P(w_1)=P(X|w_2)*P(w_2)}$$

- Applying logartihm:

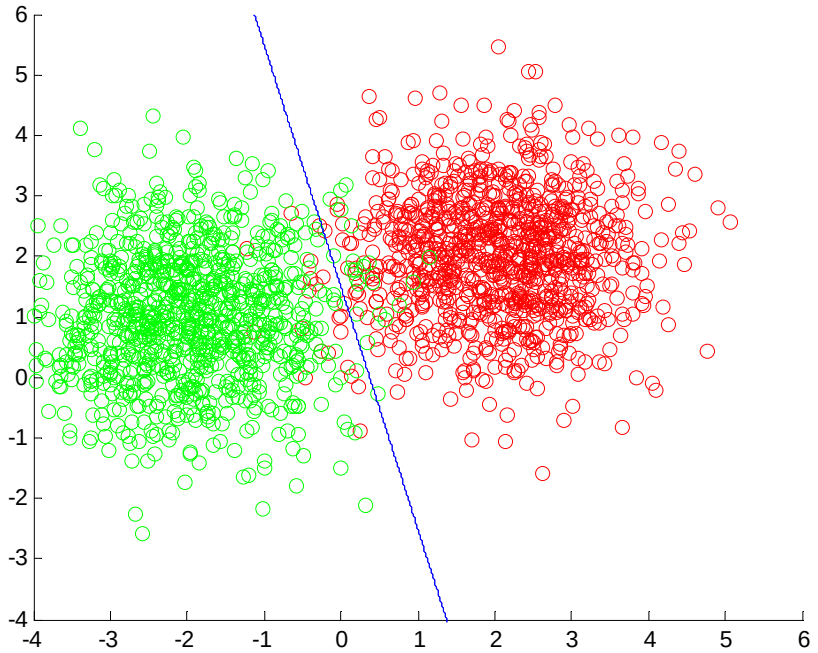
$$\ln(P(w_1)) - 0.5\cdot\ln(2\pi\cdot|C|) - \frac{(x-\mu_1)^{'}\cdot C^{-1}\cdot(x-\mu_1)}{2} = \ln(P(w_2)) - 0.5\cdot\ln(2\pi\cdot|C|) - \frac{(x-\mu_2)^{'}\cdot C^{-1}\cdot(x-\mu_2)}{2}$$

- Developing:

$$(x-\mu_1)^{'}\cdot C^{-1}\cdot(x-\mu_1) = (x-\mu_2)^{'}\cdot C^{-1}\cdot(x-\mu_2)$$

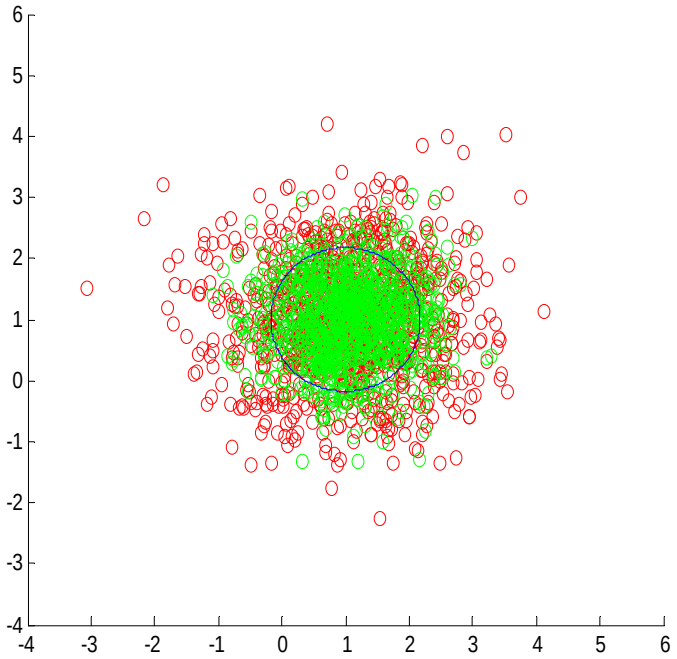
Example 1: Different means and same covariance

```
m1=[2;2]; m2=[-2;1]; C1=[1 0;0 1]; C2=[1 0;0 1];
x=[randnorm(m1,C1,1000) randnorm(m2,C2,1000)];
y=[zeros(1,1000) ones(1,1000)];
plotpat(x,y);
hold on;
plotbon(m1,C1,m2,C2,'b');
axis([-4 6 -4 6])
```



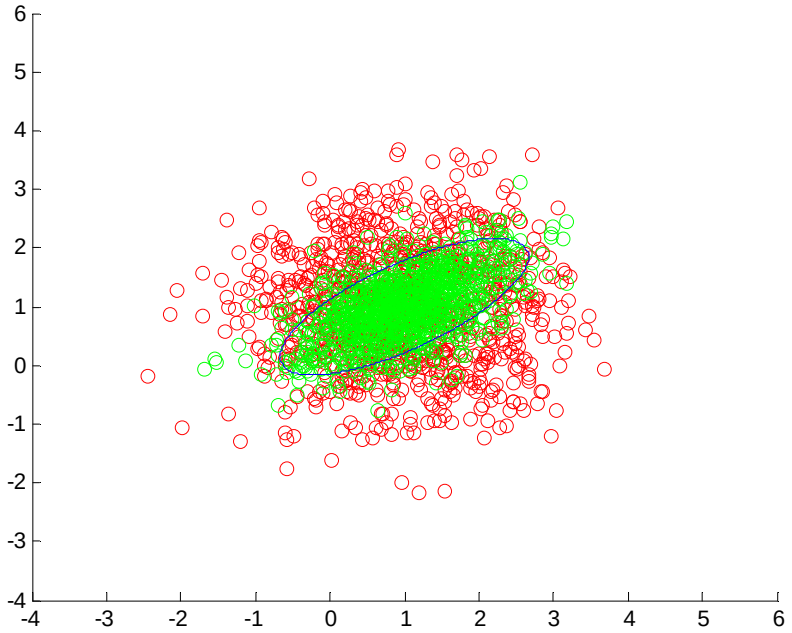
Example 2: proportional covariance and same means

```
m1=[1;1]; m2=[1;1]; C1=[1 0;0 1]; C2=[0.5 0;0 0.5];
x=[randnorm(m1,C1,1000) randnorm(m2,C2,1000)];
y=[zeros(1,1000) ones(1,1000)];
plotpat(x,y);
hold on;
plotbon(m1,C1,m2,C2,'b');
axis([-4 6 -4 6])
```



Example3: Different covariance matrix and same means

```
m1=[1;1]; m2=[1;1]; C1=[1 0;0 1]; C2=[0.5 0.2;0.2 0.3];
x=[randnorm(m1,C1,1000) randnorm(m2,C2,1000)];
y=[zeros(1,1000) ones(1,1000)];
plotpat(x,y);
hold on;
plotbon(m1,C1,m2,C2,'b');
axis([-4 6 -4 6])
```



Conclusions

- Case 1. The optimal Bayesian classifier considering classes normally distributed, equiprobables, and the same covariance matrix is the minimum distance classifier considering the Euclidean distance. In this case, the boundary is a hyper plane of dimension N-1 (a line in the 2D case)

- Case 2. When the covariance matrix varies between classes, the minimum distance classifier is obtained by considering the Mahalanobis distance. In this case, the border is a hyper quadratic (a conic in the 2D case)

CLASSIFIER OF MINIMUM PREDICTION RISK

Loss concept

- Obviously, the cost of an error in a classification problem can be very different depending on the way in which it is committed.
  - Example: In the case of deciding whether a patient has cancer or not, depending on a number of symptoms, is much more serious to commit a false positive, that a false negative
- The loss function is defined as a matrix, where the element a_{ij} is the cost of choosing the class *j* when the actual class is the *i*

$$L_{ij} = L(w_i, w_j)$$

- The most common loss function is the "zero-loss" called, where $L_{ij} = 0$ if $i = j$ and 1 otherwise

Prediction risk

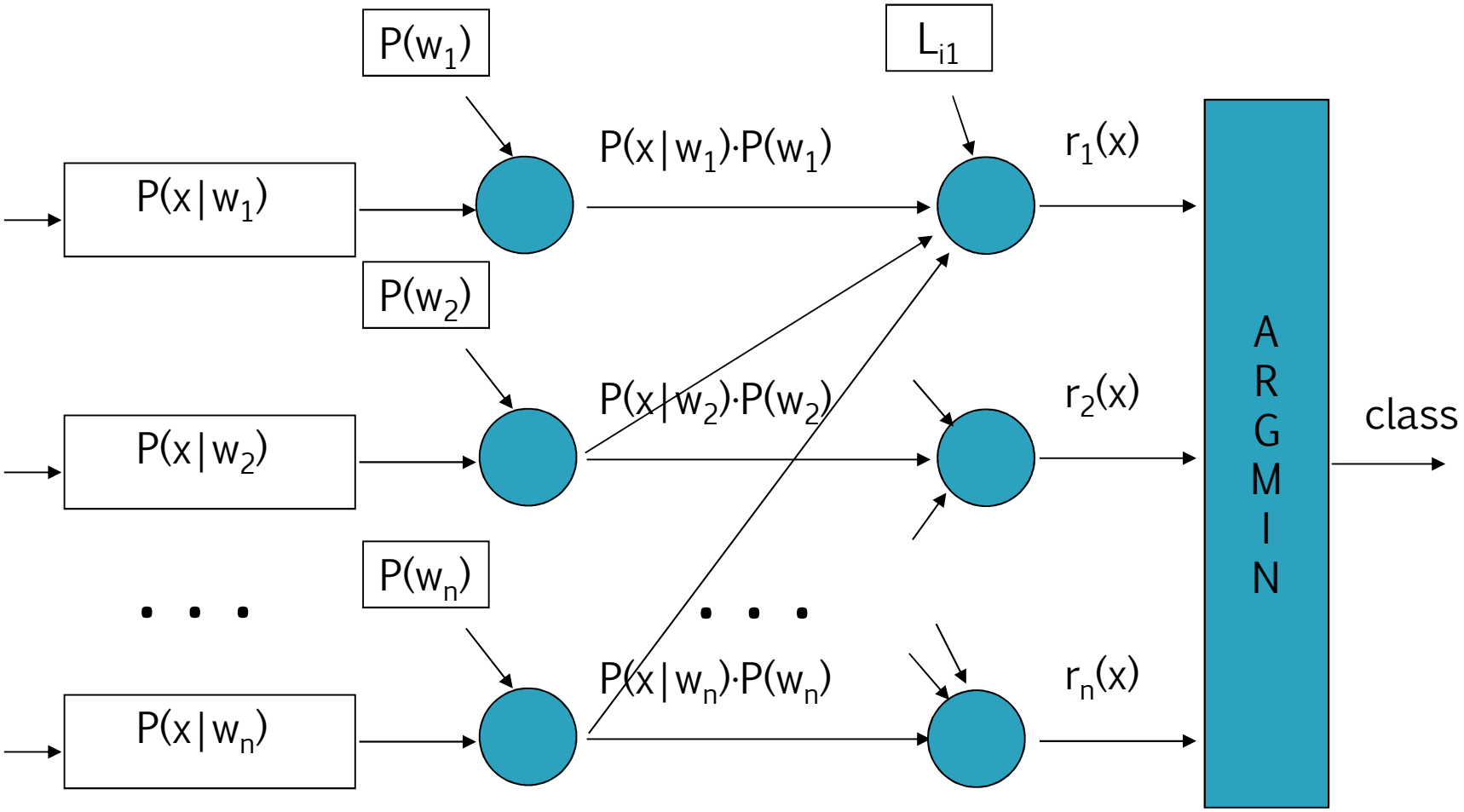
- Since a given pattern *x* can belong to any of the *M* possible classes, the average risk to assign *x* to class *j* is defined as:

$$r_j(x) = \sum_{i=1}^{M} L_{ij} \cdot p(w_i \mid x)$$

- By Bayes theorem, and discarding the term P (x):

$$r_j(x) = \sum_{i=1}^{M} L_{ij} \cdot p(x \mid w_i) \cdot p(w_i)$$

Naive Bayes classifier based on the minimum risk prediction



Loss function effect