EE 610, ST: ML Fundamentals

Bayes Classifiers

Dr. W. David Pan

Dept. of ECE

UAH

Topics

- Minimum Distance Classifier
- Optimal Bayes Classifier
- Maximum Likelihood Estimation of parameters
- Naïve Bayes Classifier

Prototype Matching

- Minimum Distance Classifier
  - Compute a distance-based measure between an unknown pattern vector and each of the class prototypes.
  - The prototype vectors are the mean vectors of the various pattern classes

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}_j \qquad j = 1, 2, \ldots, W$$

$$D_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_j\| \qquad j = 1, 2, \ldots, W$$

$$\|\mathbf{a}\| = (\mathbf{a}^T \mathbf{a})^{1/2}$$ is the Euclidean Norm

  - Then assign the unknown pattern to the class of its closest prototype.

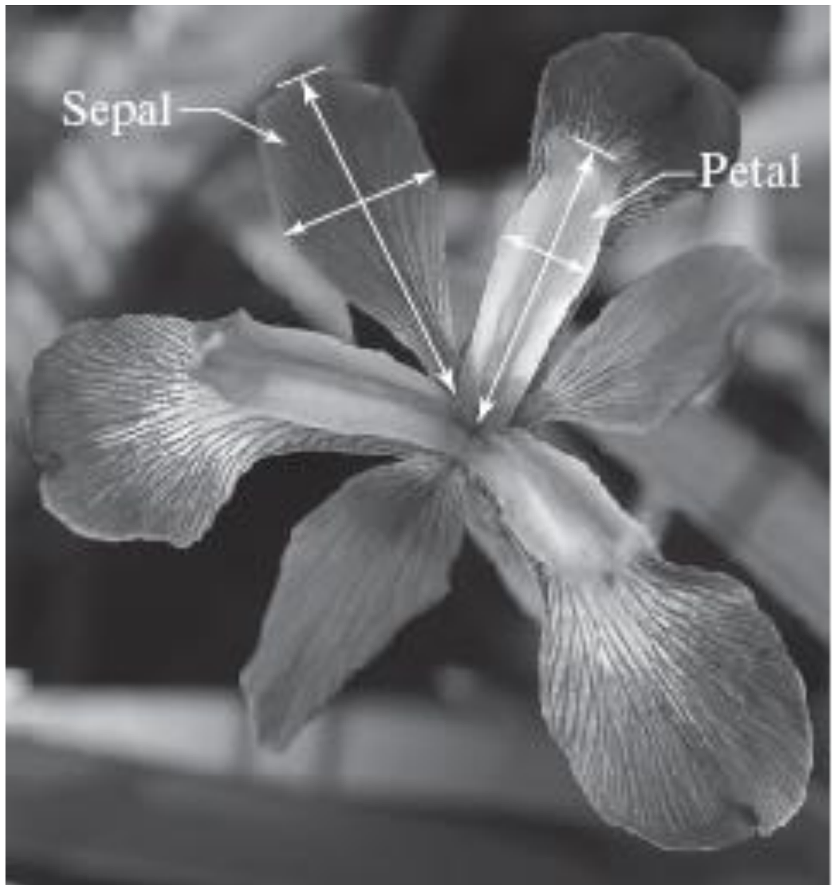
- It can be shown that it is equivalent to selecting a class that can maximize the following decision function:

$$d_j(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T \mathbf{m}_j \qquad j = 1, 2, \ldots, W$$

- The decision boundary between two classes:

$$\begin{aligned} d_{ij}(\mathbf{x}) &= d_i(\mathbf{x}) - d_j(\mathbf{x}) \\ &= \mathbf{x}^T(\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2}(\mathbf{m}_i - \mathbf{m}_j)^T(\mathbf{m}_i + \mathbf{m}_j) = 0 \end{aligned}$$

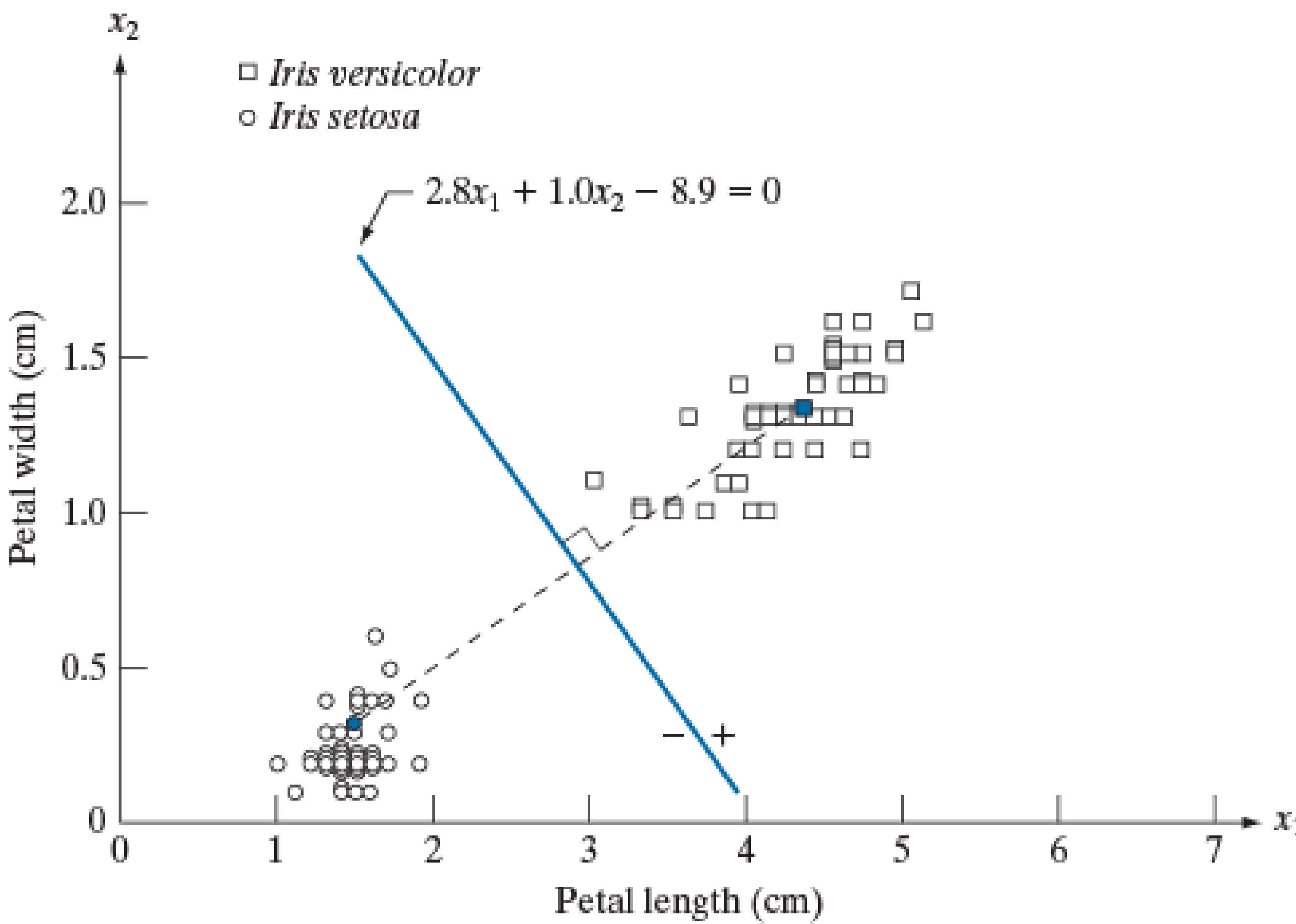
Feature Vector of Iris Dataset



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

x_1 = Petal width
x_2 = Petal length
x_3 = Sepal width
x_4 = Sepal length

Illustration for Two Classes



Detailed Derivations

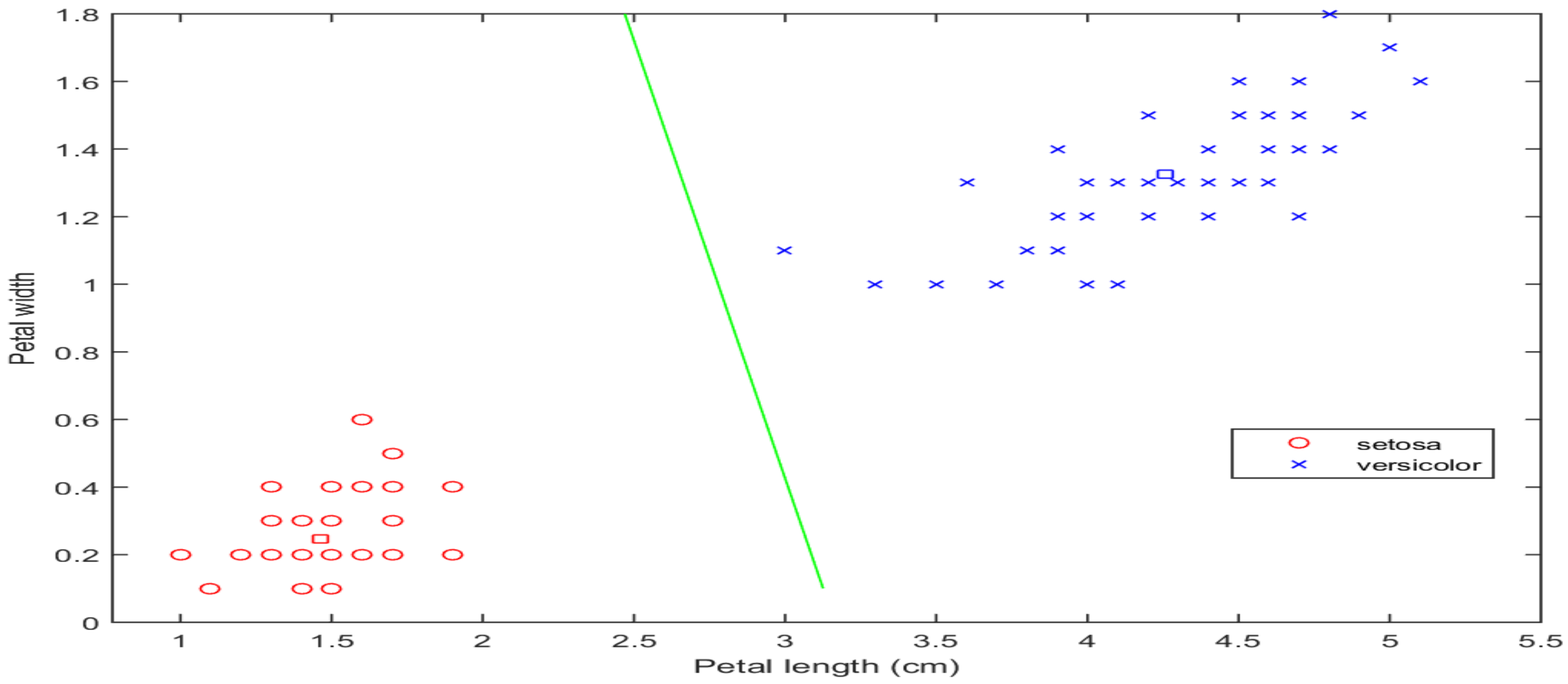
$$\mathbf{m}_1 = (4.3, 1.3)^T \quad \mathbf{m}_2 = (1.5, 0.3)^T.$$

$$\begin{aligned} d_1(\mathbf{x}) &= \mathbf{x}^T\mathbf{m}_1 - \frac{1}{2}\mathbf{m}_1^T\mathbf{m}_1 \\ &= 4.3x_1 + 1.3x_2 - 10.1 \end{aligned}$$

$$\begin{aligned} d_2(\mathbf{x}) &= \mathbf{x}^T\mathbf{m}_2 - \frac{1}{2}\mathbf{m}_2^T\mathbf{m}_2 \\ &= 1.5x_1 + 0.3x_2 - 1.17 \end{aligned}$$

$$\begin{aligned} d_{12}(\mathbf{x}) &= d_1(\mathbf{x}) - d_2(\mathbf{x}) \\ &= 2.8x_1 + 1.0x_2 - 8.9 = 0 \end{aligned}$$

Matlab

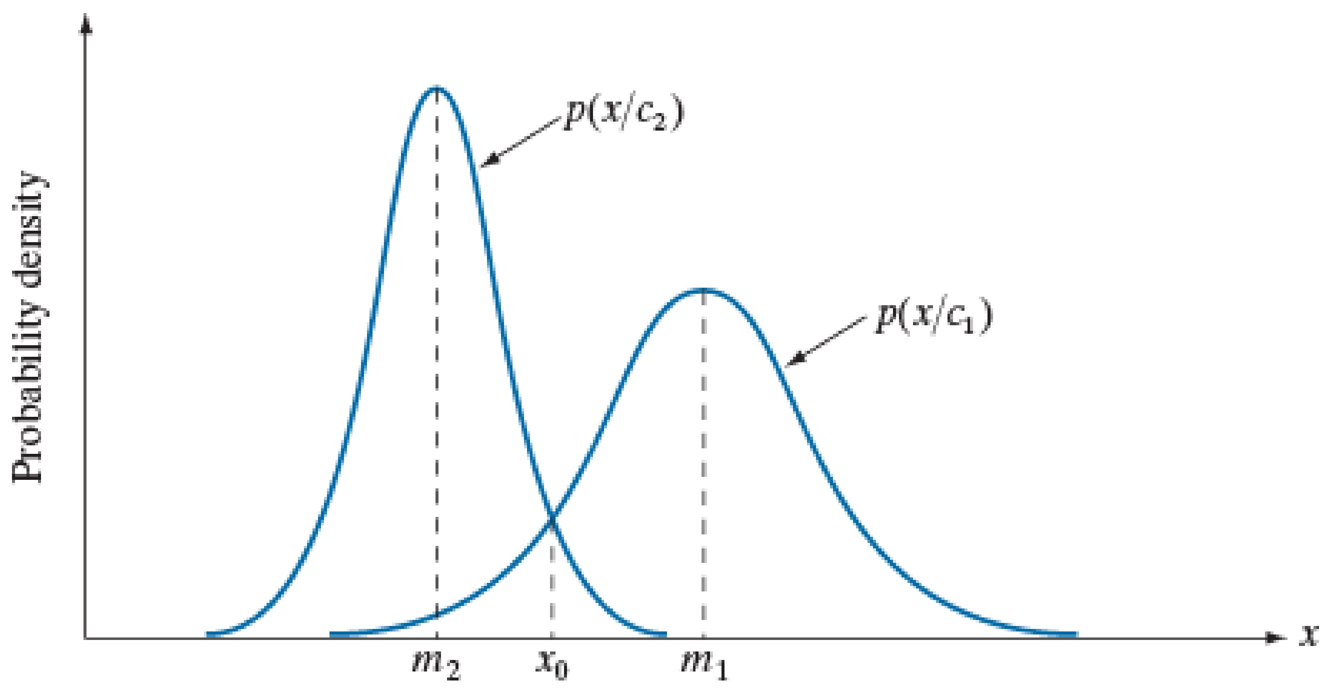


1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Petal width
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
Petal length (cm)
setosa
versicolor

Optimal Bayes Classifier

Optimal Classification

- Probability considerations become important in pattern recognition because of the randomness under which pattern classes normally are generated.
- It is possible to derive a classification approach that is optimal in the sense that, on average, it yields the lowest probability of committing classification errors.



Conditional Probabilities and Bayes Theorem

- Joint Probability $P(A, B)$ for random events A and B.
- Conditional Probability $P(A|B) = \frac{P(A,B)}{P(B)}$. Similarly, $P(B|A) = \frac{P(A,B)}{P(A)}$
- If events A and B are independent, then $P(A, B) = P(A)P(B)$, implying that $P(B|A) = P(B)$ and $P(A|B) = P(A)$
- Example: Ice Cream
  70% of your friends like Chocolate, and 35% like Chocolate AND like Strawberry.
  Question: What percent of those who like Chocolate also like Strawberry?

  Answer:
  P(S|C) = P(C, S) / P(C) = 0.35/0.7 = 50%

Example

A noisy communication channel modeled by transition probabilities:

Given:

Binary source: $P(S0) + P(S1) = 1$

and the ***a priori*** probabilities: $P(R0|S0) + P(R1|S0) = 1$, $P(R0|S1) + P(R1|S1) = 1$

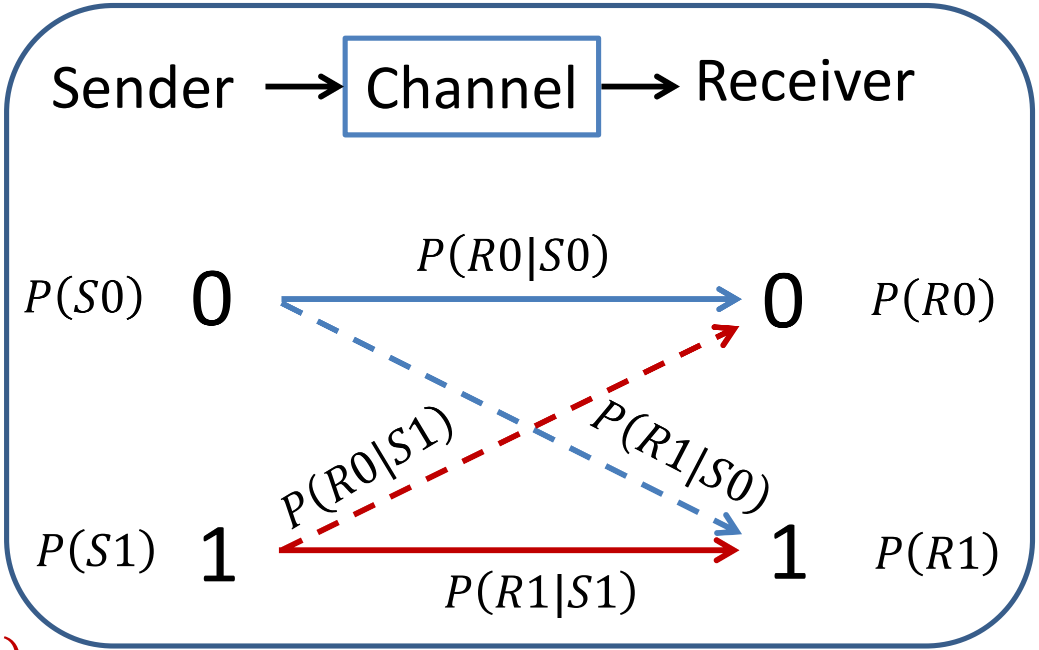
Question:

Determine $P(R0)$, $P(R1)$, and ***posterior*** probabilities $P(S0/R0)$, $P(S1/R1)$?

Answer:

$$P(R0)$$
$$= P(R0, S0) + P(R0, S1)$$
$$= P(R0|S0)P(S0) + P(R0|S1)P(S1)$$

$$P(S0|R0)$$
$$= \frac{P(R0, S0)}{P(R0)} = \frac{P(R0|S0)P(S0)}{P(R0)}$$



Decision, given the same $P(R0)$:

Accept $R0$ if $P(S0|R0) > P(S1|R0)$,

or $P(R0|S0)P(S0) > P(R0|S1)P(S1)$

Bayes Classifier

- Given the prob. that a pattern vector $\boldsymbol{x}$ comes from class c_i is denoted by $p(c_i|x)$.
- If the pattern classifier decides that $\boldsymbol{x}$ came from class c_j when it actually came from c_i, it incurs a loss denoted by L_{ij}.
- Because the pattern vector $\boldsymbol{x}$ may belong to any one of N possible classes, the average loss incurred in assigning to class c_j is

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} L_{kj} p(c_k|\boldsymbol{x})$$

which is called the *conditional average risk* in decision theory.

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} L_{kj} p(c_k|\boldsymbol{x})$$

According to the Bayes Theorem

$$p(c_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|c_k)P(c_k)}{p(\boldsymbol{x})},$$

Therefore,

$$r_j(\boldsymbol{x}) = \frac{1}{p(\boldsymbol{x})} \sum_{k=1}^{N} L_{kj} p(\boldsymbol{x}|c_k) P(c_k)$$

where

$p(\boldsymbol{x}|c_k)$: PDF of the patterns from class c_k;
(*a priori* prob.)

$P(c_k)$: Prob. of occurrence of class c_k

Since $p(\boldsymbol{x})$ is a common term, we can rewrite $r_j(\boldsymbol{x})$ as

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} L_{kj} p(\boldsymbol{x}|c_k) P(c_k)$$

The classifier that minimizes the total average loss Is called the **Bayes Classifier**,

where the classifier assigns an unknown pattern $\boldsymbol{x}$ to class c_i if $r_i(\boldsymbol{x}) < r_j(\boldsymbol{x})$ for $j = 1, 2, \ldots, N; j \neq i$. That is

$$\sum_{k=1}^{N} L_{ki} p(\boldsymbol{x}|c_k) P(c_k) < \sum_{q=1}^{N} L_{qj} p\left(\boldsymbol{x}\middle|c_q\right) P\left(c_q\right)$$

If the loss for a correct decision is generally assigned a value of 0, and the loss for an incorrect decision is assigned a value of 1, then $L_{ij} = 1 - \delta_{ij}$.

Derivation of the Bayes Classifier

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} L_{kj} p(\boldsymbol{x}|c_k) P(c_k) \quad \text{and} \quad L_{kj} = 1 - \delta_{kj}$$

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} (1 - \delta_{ij}) p(\boldsymbol{x}|c_k) P(c_k)$$

$$= \sum_{k=1}^{N} p(\boldsymbol{x}|c_k) P(c_k) - \sum_{k=1}^{N} \delta_{ij} p(\boldsymbol{x}|c_k) P(c_k)$$

$$= p(\boldsymbol{x}) - p\left(\boldsymbol{x}|c_j\right) P\left(c_j\right)$$

Similarly,

$$r_i(\boldsymbol{x}) = p(\boldsymbol{x}) - p(\boldsymbol{x}|c_i) P(c_i)$$

Decision Rule

- classifier assigns an unknown pattern $\boldsymbol{x}$ to class c_i if
  $r_i(\boldsymbol{x}) < r_j(\boldsymbol{x})$ for $j = 1, 2, \dots, N; j \neq i$.

$$p(\boldsymbol{x}) - p(\boldsymbol{x}|c_i)P(c_i) < p(\boldsymbol{x}) - p(\boldsymbol{x}|c_j)P(c_j),$$

or equivalently

$$\boxed{p(\boldsymbol{x}|c_i)P(c_i) > p(\boldsymbol{x}|c_j)P(c_j)}$$

Decision Function

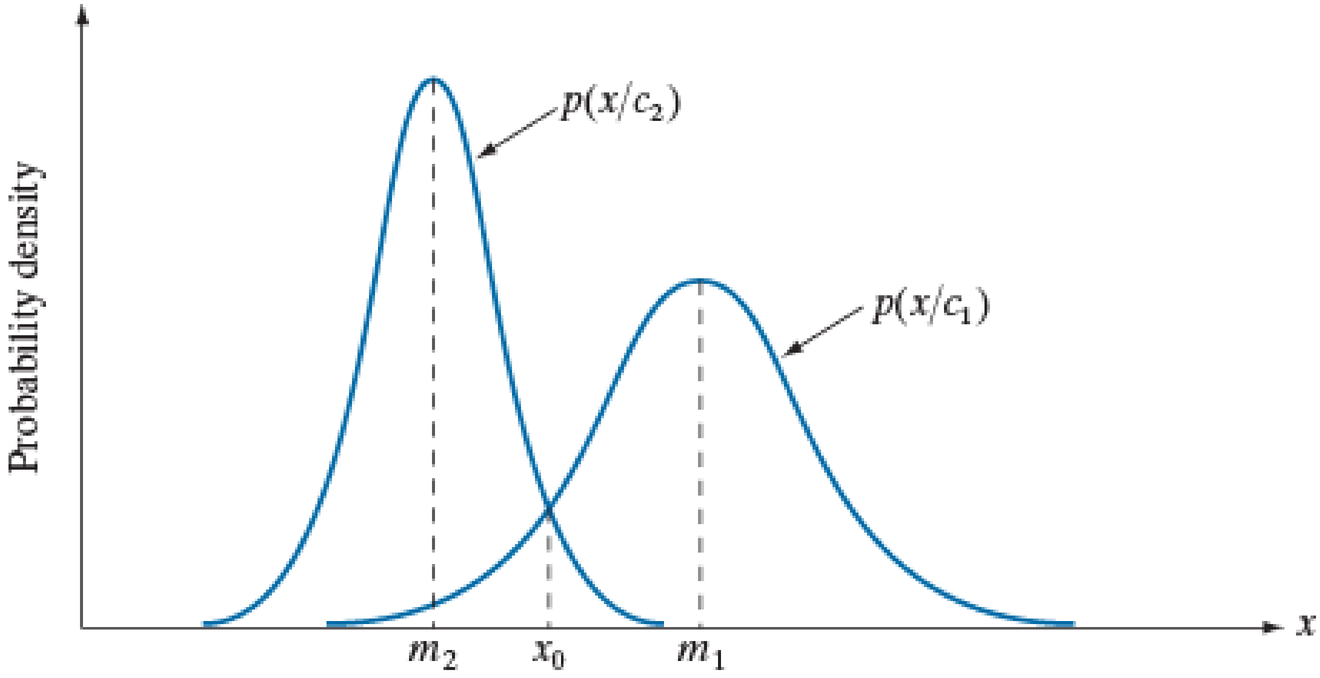
- The Bayes Classifier for a 0-1 loss function computes the decision function

$$d_j(\boldsymbol{x}) = p(\boldsymbol{x}|c_i)P(c_i)$$

for $j = 1, 2, \ldots, N$ and assign a pattern $\boldsymbol{x}$ to class c_i if $d_i(\boldsymbol{x}) > d_j(\boldsymbol{x})$, for all $j \neq i$.

- For the optimality of Bayes decision function to hold, the *a priori* probability $p(\boldsymbol{x}|c_i)$ and the class probability $P(c_i)$ needs to be known or estimated from sample patterns during training.
- Usually assume Gaussian Distribution for $p(\boldsymbol{x}|c_i)$.

Gaussian Pattern Classes



$$d_j(x) = p(x|c_j)P(c_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(c_j)$$

where $j = 1, 2$

n-Dimensional Gaussian PDF

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T \mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}$$

where the mean vector is $\quad \mathbf{m}_j = E_j\{\mathbf{x}\}$

and the covariance matrix is

$$\mathbf{C}_j = E_j\{(\mathbf{x} - \mathbf{m}_j)(\mathbf{x} - \mathbf{m}_j)^T\}$$

We can approximate with taking the averages of sample vectors:

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x} \qquad \mathbf{C}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}\mathbf{x}^T - \mathbf{m}_j \mathbf{m}_j^T$$

Logarithm of the Decision Function

$$d_j(\mathbf{x}) = \ln\left[p(\mathbf{x}/\omega_j)P(\omega_j)\right] = \ln\, p(\mathbf{x}/\omega_j) + \ln\, P(\omega_j)$$

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}$$

$$d_j(\mathbf{x}) = \ln\, P(\omega_j) - \frac{n}{2}\ln 2\pi - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}\left[(\mathbf{x} - \mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x} - \mathbf{m}_j)\right]$$

$$d_j(\mathbf{x}) = \ln\, P(\omega_j) - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}\left[(\mathbf{x} - \mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x} - \mathbf{m}_j)\right]$$

- If the covariance matrix is identical. then

$$d_j(\mathbf{x}) = \ln P(\omega_j) + \mathbf{x}^T\mathbf{C}^{-1}\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{C}^{-1}\mathbf{m}_j$$

- If all classes are equally likely and the covariance matrix is an identity matrix, then

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \qquad j = 1, 2, \ldots, W$$

- The same decision function for a minimum-distance classifier, which is optimal in the Bayes sense if
  - The pattern classes are Gaussian.
  - All covariance matrices are equal to identity matrix.
  - All classes are equally likely.

Example

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad m_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, m_2 = \begin{bmatrix} 9 \\ 9 \end{bmatrix},$$

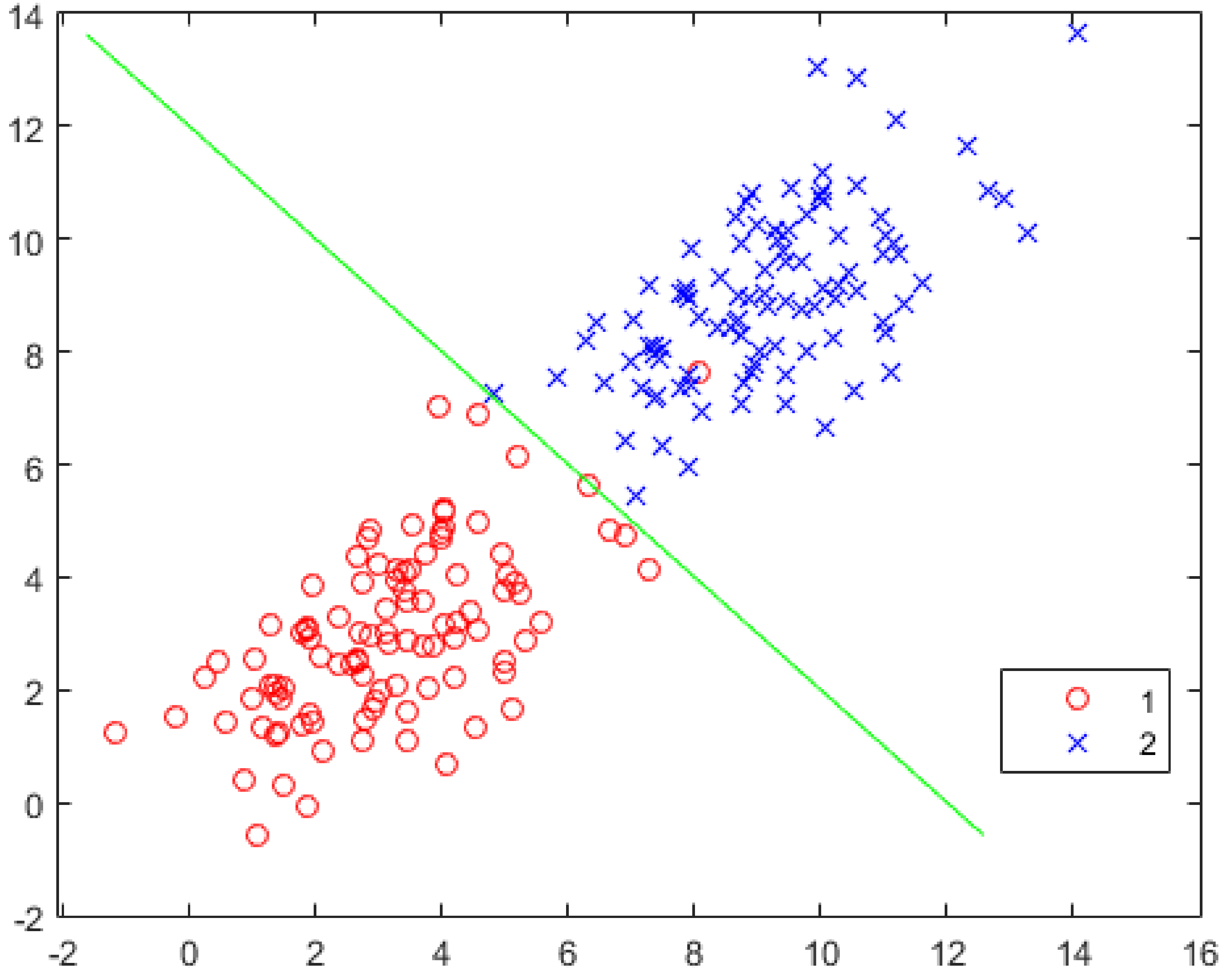
$$C_1 = C_2 = C = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \qquad C^{-1} = \frac{1}{3}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

$$d_j(\boldsymbol{x}) = \boldsymbol{x}^T C^{-1} m_j - \frac{1}{2} m_j^T C^{-1} m_j$$

$d_1(\boldsymbol{x}) = x_1 + x_2 - 3$ and
$d_2(\boldsymbol{x}) = 3x_1 + 3x_2 - 27$

The decision boundary is

$$d_2(\boldsymbol{x}) - d_1(x) = x_1 + x_2 - 12 = 0$$


14
12
10
8
6
4
2
0
-2
-2
0
2
4
6
8
10
12
14
16
1
2

Parametric Form for $p(C_k|\mathbf{x})$

- Assume that the class-conditional densities are Gaussian.
- We consider first two classes, and assume that all classes share the same covariance matrix.
- Thus the density for class C_k is given by

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}_k) \right\}$$

Decision functions (with a common covariance matrix $\boldsymbol{C}$, where $\boldsymbol{C}^{\mathrm{T}} = \boldsymbol{C}$):

$$d_1(\mathbf{x}) = \ln P(\omega_1) - \frac{1}{2}\ln|\boldsymbol{C}| - \frac{1}{2}[(\mathbf{x}-\mathbf{m_1})^{\mathrm{T}}\mathrm{C}^{-1}(\mathbf{x}-\mathbf{m_1})]$$

$$d_2(\mathbf{x}) = \ln P(\omega_2) - \frac{1}{2}\ln|\boldsymbol{C}| - \frac{1}{2}[(\mathbf{x}-\mathbf{m_2})^{\mathrm{T}}\mathrm{C}^{-1}(\mathbf{x}-\mathbf{m_2})]$$

Assuming equal class probabilities:

$$d_1(\mathbf{x}) = d_2(\mathbf{x}) \quad \Rightarrow \quad (\mathbf{m_1}-\mathbf{m_2})^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{x} = \frac{1}{2}(\mathbf{m_1^{\mathrm{T}}}\mathbf{C}^{-1}\mathbf{m_1} - \mathbf{m_2^{\mathrm{T}}}\mathbf{C}^{-1}\mathbf{m_2})$$

Maximum Likelihood Estimation

- Once we have specified a parametric functional form for the class-conditional densities, we can then determine the values of the parameters, together with the prior class probabilities $p(C_k)$, using maximum likelihood.
- This requires a data set comprising observations of **x** along with their corresponding class labels.
- Consider first the case of two classes, each having a Gaussian class-conditional density with a shared covariance matrix, and suppose we have a data set $\{\mathbf{x}_n, t_n\}$, where $n = 1, \ldots, N$. Here $t_n = 1$ denotes class C_1 and $t_n = 0$ denotes class C_2.
- We denote the prior class probability $p(C_1) = \pi$, so that $p(C_2) = 1 - \pi$.
- For a data point $\mathbf{x}_n$ from class C_1, we have $t_n = 1$ and hence

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n|\mathcal{C}_1) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}).$$

- Similarly, for a data point $\mathbf{x}_n$ from class C_2, we have $t_n = 0$ and hence

$$p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\mathbf{x}_n|\mathcal{C}_2) = (1 - \pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}).$$

The likelihood function is given by

$$p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} [\pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

where $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$

It is convenient to maximize the **log** of the likelihood function.

- Consider first the maximization with respect to π.
  - The terms in the log likelihood function that depend on π are

$$\sum_{n=1}^{N} \{t_n \ln \pi + (1-t_n) \ln(1-\pi)\}$$

  - Setting the derivative with respect to π equal to zero, we obtain

$$\pi = \frac{1}{N}\sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}$$

  - Thus the maximum likelihood estimate for π is the fraction of points in class C_1 as expected. This can be generalized to the multiclass case, where the maximum likelihood estimate of the prior probability associated with class C_k is given by the fraction of the training set points assigned to that class.

.

Maximum Likelihood Estimate of the Means

- We can pick out of the log likelihood function those terms that depend on $\boldsymbol{\mu}_1$

$$\sum_{n=1}^{N} t_n \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{n=1}^{N} t_n(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_1) + \text{const.}$$

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n|\mathcal{C}_1) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}).$$

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

- Setting the derivative with respect to $\boldsymbol{\mu}_1$ to zero, we can obtain

$$\boldsymbol{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n \mathbf{x}_n$$

  which is simply the mean of all the input vectors x_n assigned to class C_1.

- By a similar argument, we have

$$\boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N}(1 - t_n)\mathbf{x}_n$$

  which is simply the mean of all the input vectors x_n assigned to class C_2.

Matrix Calculus

For a scalar α given by a quadratic form: $\alpha = \mathbf{x}^T \mathbf{A} \mathbf{x}$

where $\mathbf{x}$ *is* $n \times 1$, $\mathbf{A}$ *is* $n \times n$, *and* $\mathbf{A}$ *does not depend on* $\mathbf{x}$, *then*

$$\frac{\partial \alpha}{\partial \mathbf{x}} = \mathbf{x}^T \left(\mathbf{A} + \mathbf{A}^T\right)$$

Proof: *By definition*

$$\alpha = \sum_{j=1}^{n} \sum_{i=1}^{n} a_{ij} x_i x_j$$

Differentiating with respect to the k*th element of* $\mathbf{x}$ *we have*

$$\frac{\partial \alpha}{\partial x_k} = \sum_{j=1}^{n} a_{kj} x_j + \sum_{i=1}^{n} a_{ik} x_i$$

for all $k = 1, 2, \ldots, n$, *and consequently,*

$$\frac{\partial \alpha}{\partial \mathbf{x}} = \mathbf{x}^T \mathbf{A}^T + \mathbf{x}^T \mathbf{A} = \mathbf{x}^T \left(\mathbf{A}^T + \mathbf{A}\right)$$

For the special case $\mathbf{A}^T = \mathbf{A}$, then $\frac{\partial[\mathbf{x}^T(\mathbf{A}+\mathbf{A}^T)]}{\partial x} = 2\mathbf{x}^T\mathbf{A}$

Naïve Bayes Classifier

- Naive Bayes methods are based on applying Bayes' theorem with the "naive" assumption of conditional *independence* between every pair of features given the value of the class variable.
- Suppose $\boldsymbol{x} = (x_1, x_2, \ldots, x_\mathrm{n})$
- Therefore,

$$p(\boldsymbol{x}|c_j) = p(x_1,\ x_2, \ldots, x_\mathrm{n}|c_j) = \prod_{k=1}^{n} p(x_i|c_j)$$

2-D Gaussian Distribution with Independent Components

$f(\mathbf{x}) = \frac{1}{2\pi\sqrt{|C|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^{\mathrm{T}}\mathrm{C}^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right]$, where $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\bar{\mathbf{x}} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$, and

$\mathrm{C} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$, then $\sqrt{|C|} = \sqrt{\sigma_1^2\sigma_2^2} = \sigma_1\sigma_2$, $\mathrm{C}^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix}$

$$(\mathbf{x}-\bar{\mathbf{x}})^{\mathrm{T}}\mathrm{C}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) = \begin{bmatrix} x_1-\mu_1 & x_2-\mu_2 \end{bmatrix} \begin{bmatrix} \dfrac{1}{\sigma_1^2} & 0 \\ 0 & \dfrac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix} =$$

$$= \frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}$$

Thus

$$f(\mathbf{x}) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left\{-\frac{1}{2}\left[\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}\right]\right\} = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_1-\mu_1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_2-\mu_2)^2}{2\sigma_2^2}}$$

Decision Rule

- Recall: the optimal Bayes classifier assigns an unknown pattern $\boldsymbol{x}$ to class c_i if $r_i(\boldsymbol{x}) < r_j(\boldsymbol{x})$ for $j = 1, 2, \ldots, N; j \neq i$.

$$p(\boldsymbol{x}|c_i)P(c_i) > p(\boldsymbol{x}|c_j)P(c_j)$$

- Therefore, for Naïve Bayes classifier, the decision rule changes to:

$$\prod_{k=1}^{n} p(x_k|c_i)\, P(c_i) > \prod_{k=1}^{n} p(x_k|c_j)\, P(c_j)$$

Two Classes

Decision Boundary

$$\prod_{k=1}^{n} p(x_k|c_1)\,P(c_1) = \prod_{k=1}^{n} p(x_k|c_2)\,P(c_2)$$

If $P(c_1) = P(c_2)$

$$\prod_{k=1}^{n} p(x_k|c_1) = \prod_{k=1}^{n} p(x_k|c_2)$$

- We can estimate $p(c_i)$ and $p(x_k|c_j)$, where $p(c_i)$ is the relative frequency of class c_i in the training set.
- Different naïve Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $p(x_k|c_j)$.
- For example, Gaussian Naïve Bayes classifier assumes the likelihood of the features as follows (with the mean and variance being estimated from the training data).

$$p(x_k|c_j) = \frac{1}{\sqrt{2\pi\sigma_{kj}^2}} \exp\left[-\frac{(x_k - \mu_{kj})^2}{2\sigma_{kj}^2}\right]$$

Example

$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $m_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$, $m_2 = \begin{bmatrix} 9 \\ 9 \end{bmatrix}$, $C_1 = C_2 = C = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$, $C^{-1} = \frac{1}{4}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$

$$d_j(\boldsymbol{x}) = \boldsymbol{x}^T C^{-1} m_j - \frac{1}{2} m_j^T C^{-1} m_j$$

$d_1(\boldsymbol{x}) = \frac{3}{2}(x_1 + x_2 - 3)$ and

$d_2(\boldsymbol{x}) = \frac{9}{2}(x_1 + x_2 - 9)$

The decision boundary (based on optimal Bayes classifier) is

$$d_2(\boldsymbol{x}) - d_1(x) = x_1 + x_2 - 12 = 0$$

Naïve Bayes Classifier

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad m_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, m_2 = \begin{bmatrix} 9 \\ 9 \end{bmatrix}, C_1 = C_2 = C = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, C^{-1} = \frac{1}{4}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

The decision boundary (based on Naïve Bayes classifier):

$$\prod_{k=1}^{2} p(x_k|\text{Class 1}) = \frac{1}{\sqrt{2\pi 2}} \exp\left[-\frac{(x_1-3)^2}{2\cdot 2}\right] \frac{1}{\sqrt{2\pi 2}} \exp\left[-\frac{(x_2-3)^2}{2\cdot 2}\right]$$

$$\prod_{k=1}^{2} p(x_k|\text{Class 2}) = \frac{1}{\sqrt{2\pi 2}} \exp\left[-\frac{(x_1-9)^2}{2\cdot 2}\right] \frac{1}{\sqrt{2\pi 2}} \exp\left[-\frac{(x_2-9)^2}{2\cdot 2}\right]$$

$$\frac{(x_1-3)^2 + (x_2-3)^2}{4} = \frac{(x_1-9)^2 + (x_2-9)^2}{4}$$

Thus

$$x_1 + x_2 - 12 = 0$$

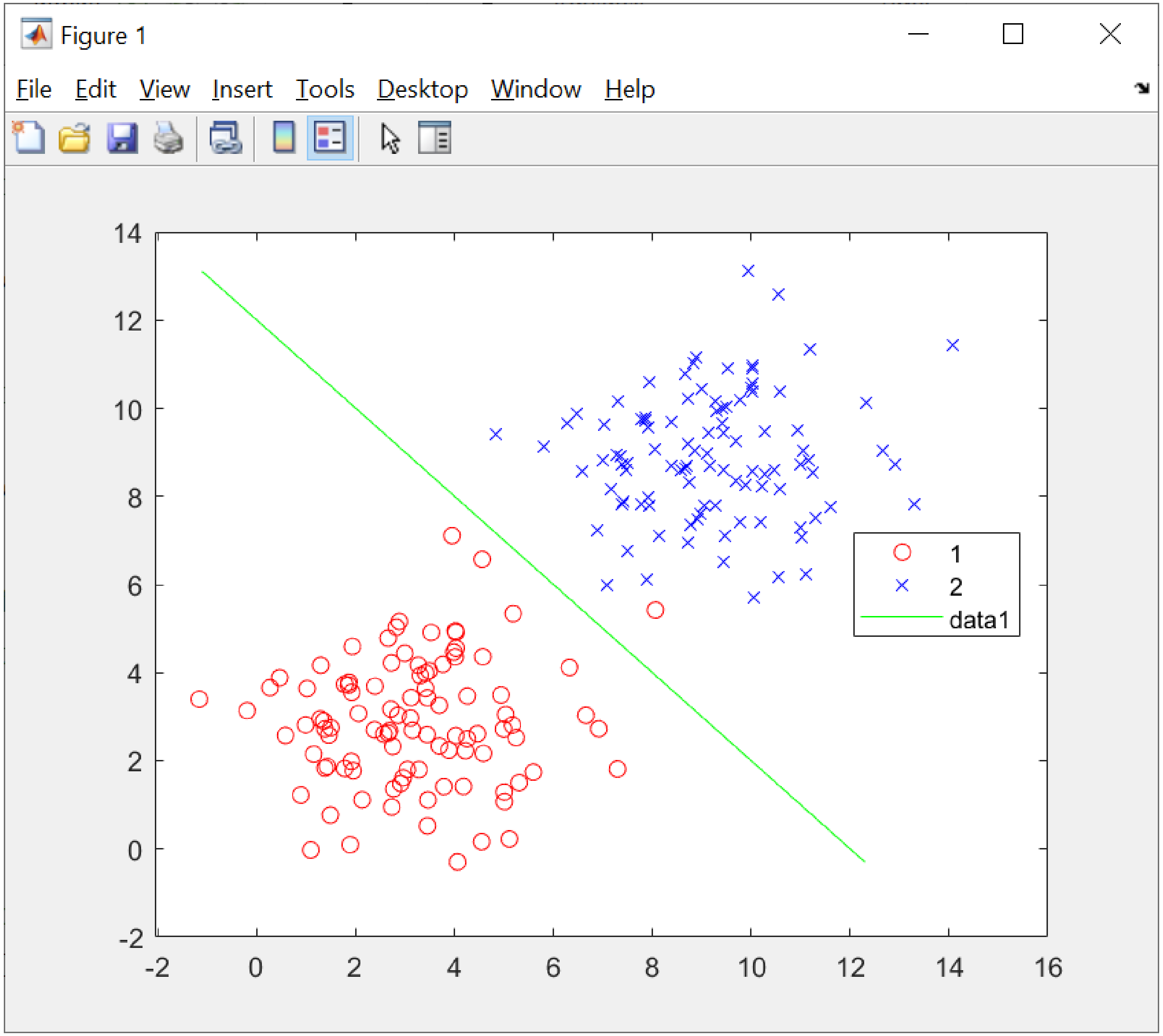


Figure 1
File Edit View Insert Tools Desktop Window Help
1
2
data1
14
12
10
8
6
4
2
0
-2
-2
0
2
4
6
8
10
12
14
16

Summary of Naïve Bayes Classifiers

- In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations (e.g., document classification and spam filtering).
- They require a small amount of training data to estimate the necessary parameters.
- The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution, which helps to alleviate the **curse of dimensionality**.