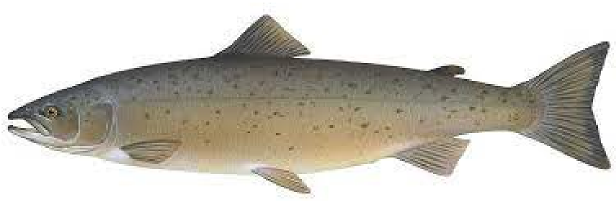
3. Bayesian Decision Theory

3.1 Bayes Theorem

Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification. We begin with the fish classification example



Let ω denote class, with $\omega = \omega_1$ for sea bass and $\omega = \omega_2$ for salmon. If the sea bass is as much as salmon, we would say the fish is equally likely to be sea bass or salmon. More generally, we assume there is a prior probability $p(\omega_1)$ that the fish is sea bass and priori probability $p(\omega_2)$ for salmon.

If there is no other types of fish, then:

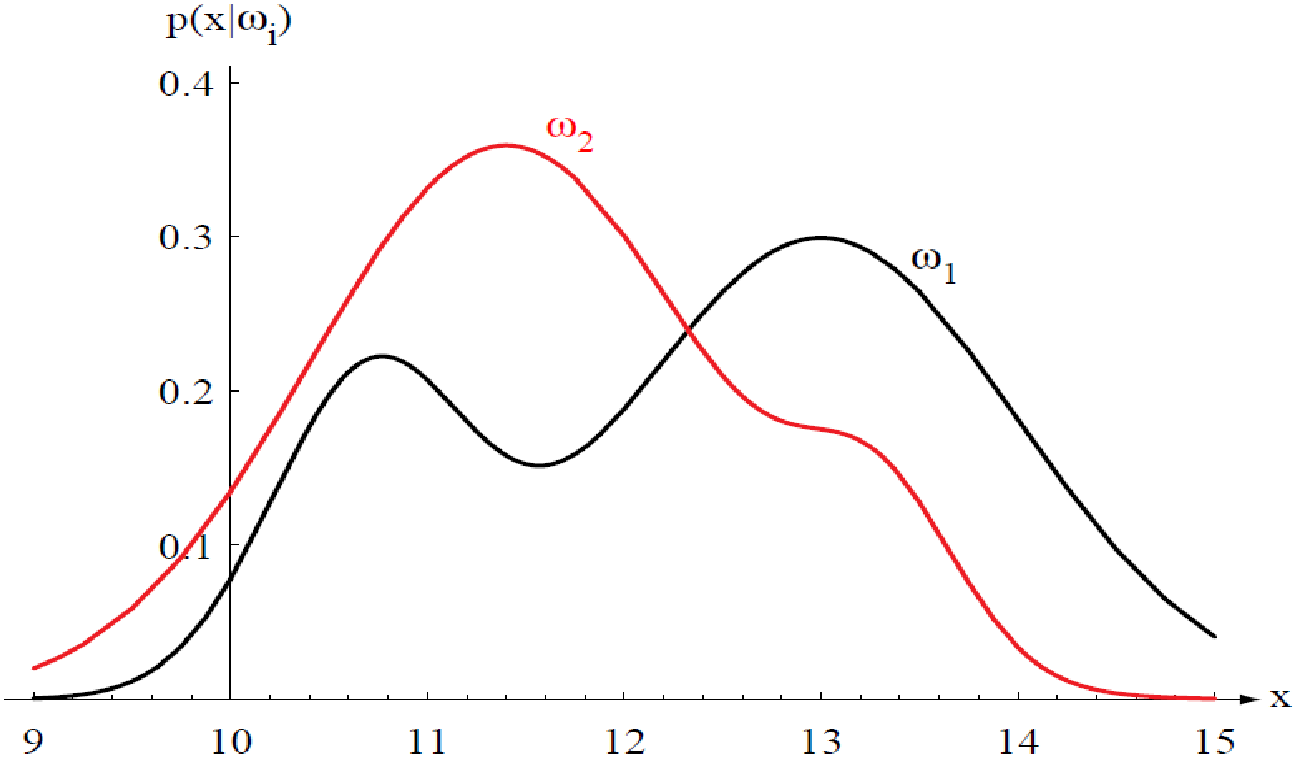
$$p(\omega_1) + p(\omega_2) = 1$$

Suppose you are not seeing the fish but is forced to make decision using the prior probability, it seems logical to use the following decision rule:

Decide ω_1 if $p(\omega_1) > p(\omega_2)$
Decide ω_2 if $p(\omega_2) > p(\omega_1)$

In most circumstances, we are given information such as the lightness measurement x to decide the type of fish. Different fish will yield different lightness measurements, this variability can be expressed in probability terms: x is considered as a continuous random variable whose distribution is expressed as $p(x|\omega)$.

$p(x|\omega)$ is the class-conditional probability density function, i.e. the probability density function for x given that the class is ω. The difference between $p(x|\omega_1)$ and $p(x|\omega_2)$ describes the difference in lightness between populations of sea bass and salmon as shown below:



Suppose we know both the prior probability and the conditional probability density. Further, we know the lightness measurement x of the fish. How this measurement could be used to decide the category of the fish?

Based on Bayes theorem:

$$p(\omega_j|x) = \frac{p(x|\omega_j)p(\omega_j)}{p(x)}$$

where

$$p(x) = \sum_{j=1}^{2} p(x|\omega_j)p(\omega_j)$$

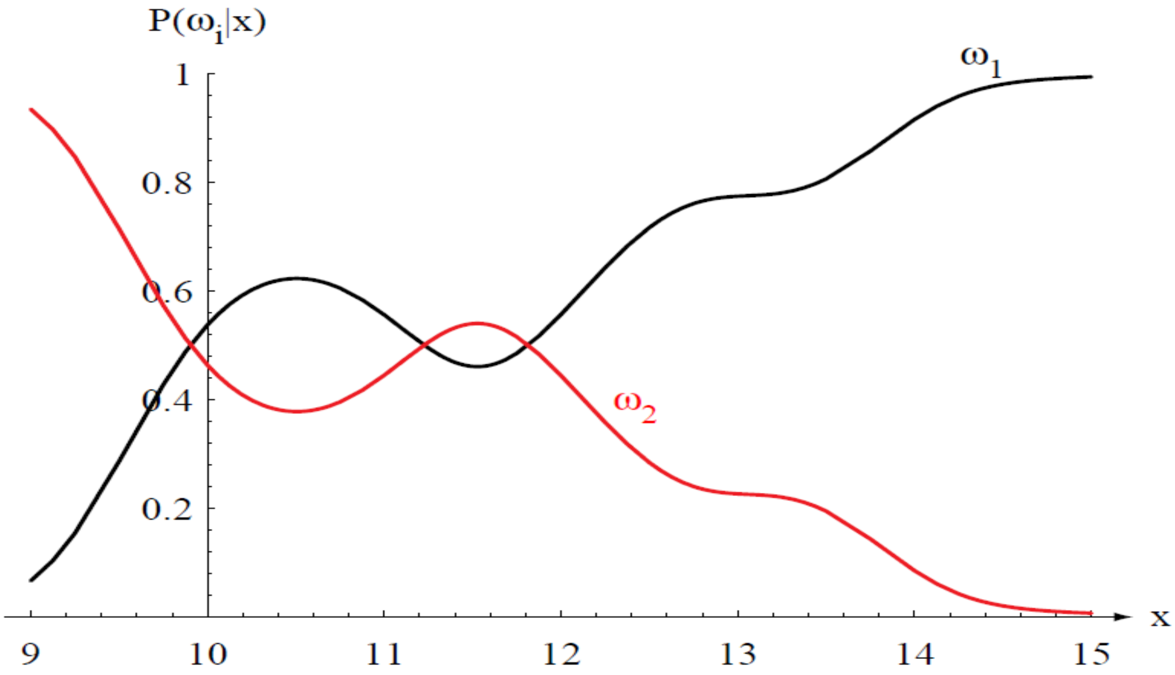
$p(\omega_j|x)$ is called posterior probability, and $p(x)$ is called evidence .

Assuming that we have the following prior probabilities:

$$p(\omega_1) = 2/3$$

$$p(\omega_2) = 1/3$$

Then the posterior probabilities are shown below:



3.2 Bayes decision rule

Decide ω_1 if $p(\omega_1|x) > p(\omega_2|x)$;

Decide ω_2 if $p(\omega_2|x) > p(\omega_1|x)$.

Note, $p(x)$ is just a scale factor and is unimportant as far as making decision is concerned. By eliminating this scale factor, we obtain the completely equivalent decision rule:

Decide ω_1 if $p(x|\omega_1)p(\omega_1) > p(x|\omega_2)p(\omega_2)$;

Decide ω_2 if $p(x|\omega_2)p(\omega_2) > p(x|\omega_1)p(\omega_1)$

We next generalize the above in 2 ways:

1) More than one feature, i.e. multiple features
2) More than two classes (categories), i.e. multiple classes

Let $\mathbf{x}$ denotes the feature vector, and $\{\omega_1, \omega_2, \cdots, \omega_c\}$ denotes the c classes. Then the posterior probability can be computed as follows:

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

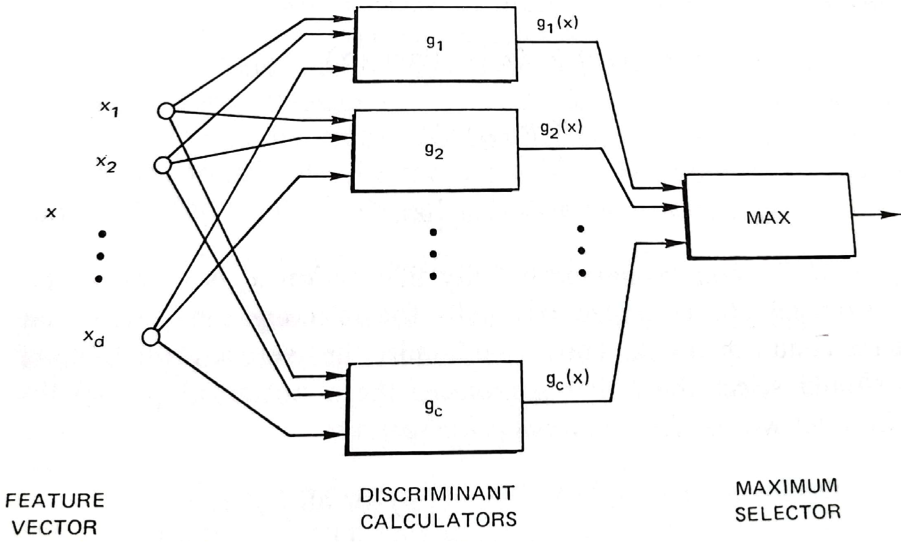
where

$$p(\mathbf{x}) = \sum_{j=1}^{c} p(\mathbf{x}|\omega_j)p(\omega_j)$$

We next introduce the concept of discriminant function in pattern classification. Assuming the discriminant functions are denoted by $g_i(\mathbf{x})$, $i = 1,2,\ldots,c$. The classifier assigns $\mathbf{x}$ to class ω_i if

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

$$i = \arg \max_{j=1,\cdots,c} g_j(\mathbf{x})$$



In Bayes decision rule, the posterior probability is used as the discriminant function:

$$g_i(\mathbf{x}) = p(\omega_i|\mathbf{x}), \qquad i = 1,2,\dots,c$$

In addition, the following variants can also be used:

$$g_i(\mathbf{x}) = p(\mathbf{x}|\omega_i)p(\omega_i)$$

$$g_i(\mathbf{x}) = \log p(\mathbf{x}|\omega_i) + \log p(\omega_i)$$

A Bayes classifier is determined by the conditional probability density function as well as the prior probability. Of the various probability density functions, the multivariate normal or Gaussian density function receives the most attentions.

3.3 Normal (Gaussian) density function

3.3.1 Univariate normal density function

$$p(x|\omega) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

Where μ is mean or the expected value of x:

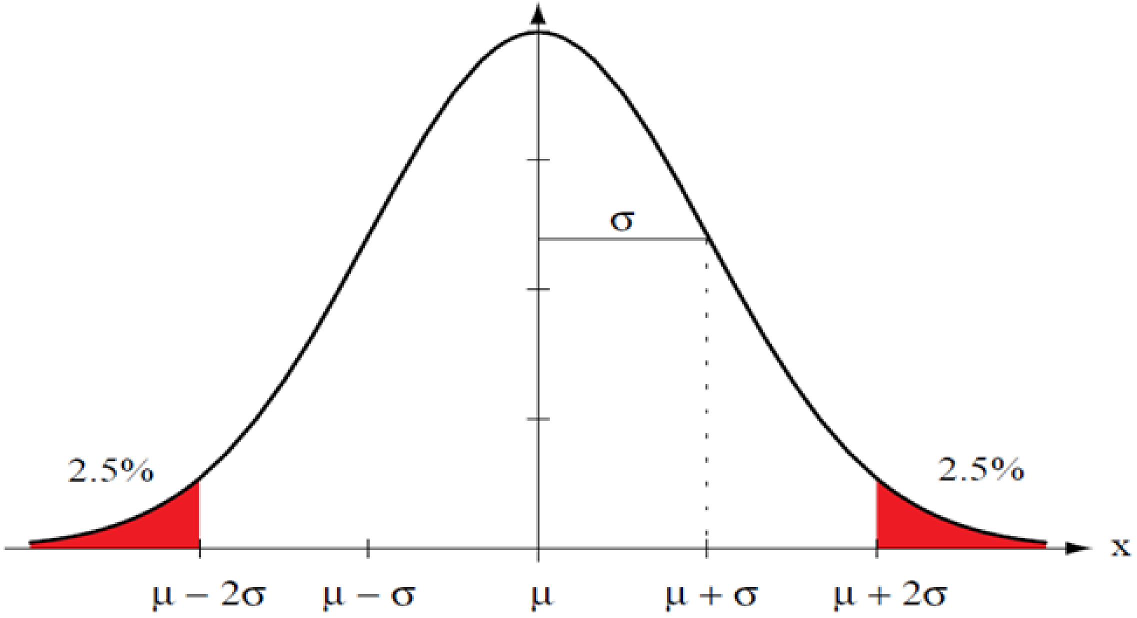
$$\mu = \int_{-\infty}^{+\infty} x p(x|\omega) dx$$

σ^2 is the variance of x:

$$\sigma^2 = \int_{-\infty}^{+\infty} (x-\mu)^2 p(x|\omega) dx$$

Normal density function is completely specified by the mean and the variance, and is often expressed as:

$$p(x|\omega) \sim N(\mu, \sigma^2)$$



3.3.2 Multivariate normal density function

The general multivariate normal density function in d dimensions is as follow:

$$p(\mathbf{x}|\omega) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})]\right]$$

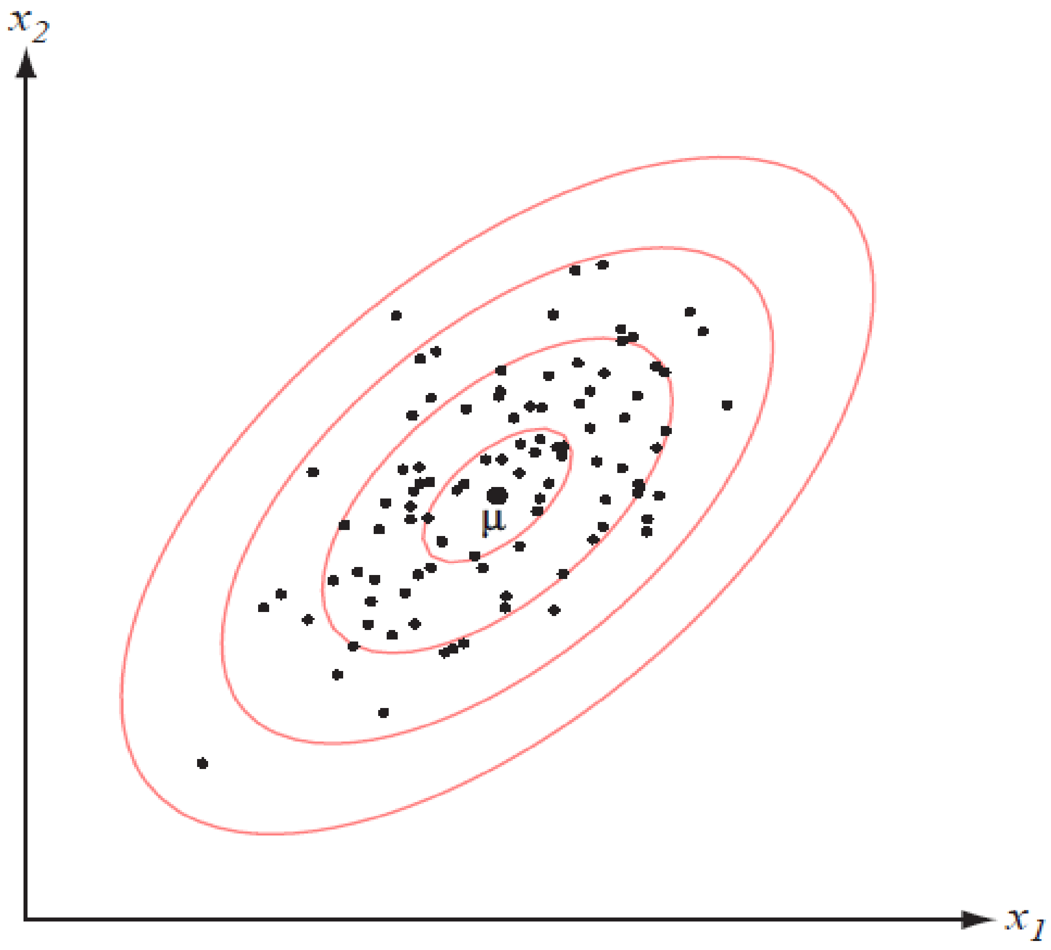
where

$\boldsymbol{\mu}$ is the d-dimensional mean vector,

$\boldsymbol{\Sigma}$ is the d-by-d covariance matrix,

$|\boldsymbol{\Sigma}|$ and $\boldsymbol{\Sigma}^{-1}$ denote the determinant and inverse of $\boldsymbol{\Sigma}$.

Samples drawn from a two-dimensional Gaussian centred at $\boldsymbol{\mu}$:



3.3.2.1 Parameter estimation

The above shows how we could design an optimal classifier if we know the priori probabilities $p(\omega_i)$ and the class-conditional densities $p(x|\omega_i)$. In practice, however, we rarely have this kind of complete knowledge about the probabilities. In a typical case, we merely have some general knowledge about the problem, together with a number of design samples or training data.

One approach to this problem is to use the samples to estimate the unknown probabilities and probability density functions, and then use the resulting estimates as if they are the true value.

$$p(\mathbf{x}|\omega) = \frac{1}{(2\pi)^{d/2}\widehat{\mathbf{\Sigma}}^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\widehat{\boldsymbol{\mu}})^T \widehat{\mathbf{\Sigma}}^{-1}(\mathbf{x}-\widehat{\boldsymbol{\mu}})]\right]$$

Where $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ denotes estimation of mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, respectively.

3.3.2.2 Maximum-likelihood parameter estimation

Suppose we have c datasets, D_1, D_2, …, D_c, with samples in D_j having been drawn independently according to the probability density function $p(\mathbf{x}|\omega_j)$. It is assumed that $p(\mathbf{x}|\omega_j) \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, and samples in D_i give no information about $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ if $i \neq j$. This permits us to work with each class separately.

With this assumption, we thus have c separate problems of the following form:

Use a set D of training samples drawn independently from the probability density $p(\mathbf{x}|\theta)$ to estimate the unknown parameter vector θ, where θ consists of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

Suppose that D contains n samples: $\mathbf{x}_1$, $\mathbf{x}_2$,…, $\mathbf{x}_n$. Because the samples are drawn independently, we have:

$$p(D|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\theta})$$

$p(D|\boldsymbol{\theta})$ is called the likelihood of $\boldsymbol{\theta}$ with respect to the set of samples D. The maximum-likelihood estimate of $\boldsymbol{\theta}$ is, by definition, the value of $\widehat{\boldsymbol{\theta}}$ that maximizes $p(D|\boldsymbol{\theta})$.

It is usually easier to work with the logarithm of the likelihood than with the likelihood itself.

We define the log-likelihood function as:

$$l(\boldsymbol{\theta}) = \ln p(D|\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln p(\mathbf{x}_k|\boldsymbol{\theta})$$

Then the solution can be written as:

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$$

Because the logarithm is monotonically increasing, the $\widehat{\boldsymbol{\theta}}$ that maximizes the log-likelihood $l(\boldsymbol{\theta})$ also maximizes the likelihood $p(D|\boldsymbol{\theta})$.

The set of necessary conditions for the maximum-likelihood estimate for $\boldsymbol{\theta}$ is as follow:

$$\frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{k=1}^{n} \frac{\partial \ln p(\mathbf{x}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$$

Case 1: Gaussian function with unknown $\boldsymbol{\mu}$

$$p(\mathbf{x}_k|\boldsymbol{\mu}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \boldsymbol{\mu})]\right]$$

Then:

$$\ln p(\mathbf{x}_k|\boldsymbol{\mu}) = -\frac{1}{2}\ln[(2\pi)^d|\boldsymbol{\Sigma}|] - \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

where $\ln$ denotes natural logarithm.

Thus, we have:

$$\frac{\partial \ln p(\mathbf{x}_k|\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = \boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

In terms of the set of necessary conditions for the maximum-likelihood estimate, we have:

$$\sum_{k=1}^{n} \boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \boldsymbol{\mu}) = 0$$

Multiplying $\boldsymbol{\Sigma}$ and rearranging, we obtain the maximum-likelihood estimate of $\boldsymbol{\mu}$:

$$\widehat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$$

Case 2: Gaussian function with unknown $\boldsymbol{\mu}$ *and* $\boldsymbol{\Sigma}$

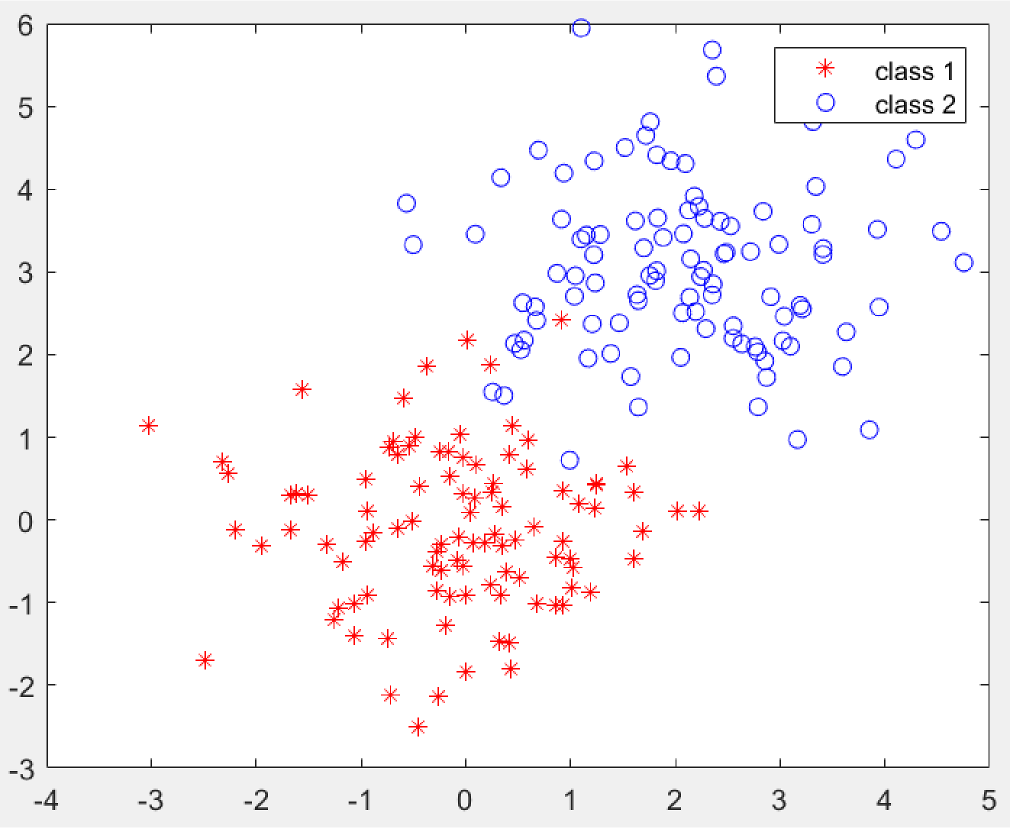
With a similar deviation process, the maximum-likelihood estimate of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are obtained as follows:

$$\widehat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{k=1}^{n} (\mathbf{x}_k - \widehat{\boldsymbol{\mu}})(\mathbf{x}_k - \widehat{\boldsymbol{\mu}})^T$$

Example:

There are 200 training samples from two classes as shown below. Assuming the data follows normal distribution, design a Bayesian decision rule using the training data.



We first estimate the parameters of the multivariate normal probability density function:

$$\widehat{\boldsymbol{\mu}}_1 = \frac{1}{n_1}\sum\nolimits_{\mathbf{x}_k \in D_1} \mathbf{x}_k = \begin{bmatrix} -0.1055 \\ -0.0974 \end{bmatrix}$$

$$\widehat{\boldsymbol{\mu}}_2 = \frac{1}{n_2}\sum\nolimits_{\mathbf{x}_k \in D_2} \mathbf{x}_k = \begin{bmatrix} 2.0638 \\ 3.0451 \end{bmatrix}$$

$$\widehat{\boldsymbol{\Sigma}}_\mathbf{1} = \frac{1}{n_1}\sum_{\mathbf{x}_k \in D_1} (\mathbf{x}_k - \widehat{\boldsymbol{\mu}}_1)(\mathbf{x}_k - \widehat{\boldsymbol{\mu}}_1)^T = \begin{bmatrix} 1.0253 & -0.0036 \\ -0.0036 & 0.8880 \end{bmatrix}$$

$$\widehat{\boldsymbol{\Sigma}}_\mathbf{2} = \frac{1}{n_2}\sum_{\mathbf{x}_k \in D_2} (\mathbf{x}_k - \widehat{\boldsymbol{\mu}}_2)(\mathbf{x}_k - \widehat{\boldsymbol{\mu}}_2)^T = \begin{bmatrix} 1.1884 & -0.013 \\ -0.013 & 1.0198 \end{bmatrix}$$

The class-conditional probability densities are then obtained as:

$$p(\mathbf{x}|\omega_1) = \frac{1}{2\pi\sqrt{|\mathbf{\Sigma}_1|}} \exp\left[-\frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_1)^T \hat{\mathbf{\Sigma}}_1^{-1} (\mathbf{x} - \hat{\mathbf{u}}_1)\right]$$

$$p(\mathbf{x}|\omega_2) = \frac{1}{2\pi\sqrt{|\mathbf{\Sigma}_2|}} \exp\left[-\frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_2)^T \hat{\mathbf{\Sigma}}_2^{-1} (\mathbf{x} - \hat{\mathbf{u}}_2)\right]$$

The discriminant functions are:

$$g_1(\mathbf{x}) = p(\mathbf{x}|\omega_1)p(\omega_1) = \frac{1}{2}p(\mathbf{x}|\omega_1)$$

$$g_2(\mathbf{x}) = p(\mathbf{x}|\omega_2)p(\omega_2) = \frac{1}{2}p(\mathbf{x}|\omega_2)$$

For any given sample $\mathbf{x}$,

1) It is classified into class 1 if

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

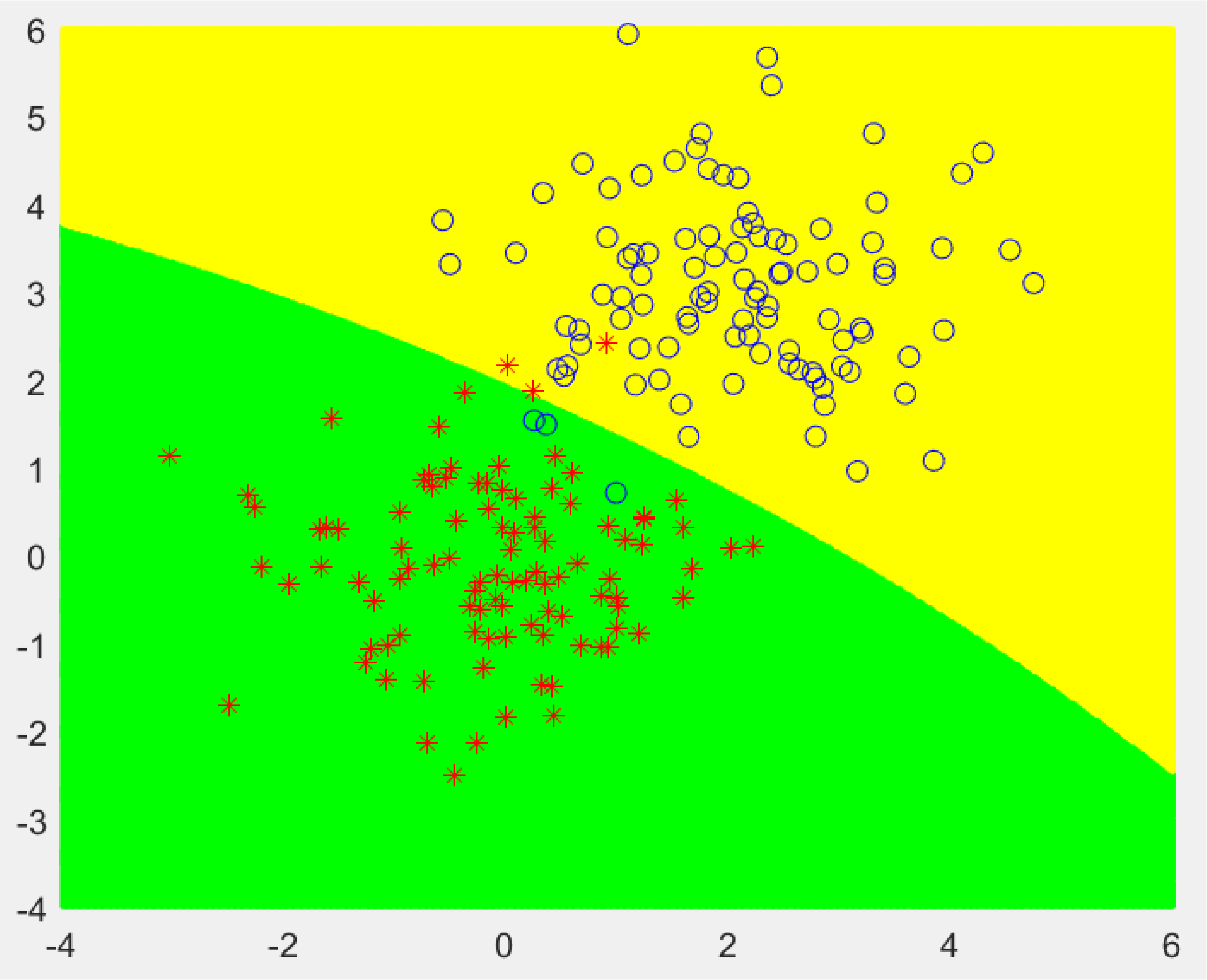
2) It is classified into class 2 if

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

3) It is on the decision boundary if

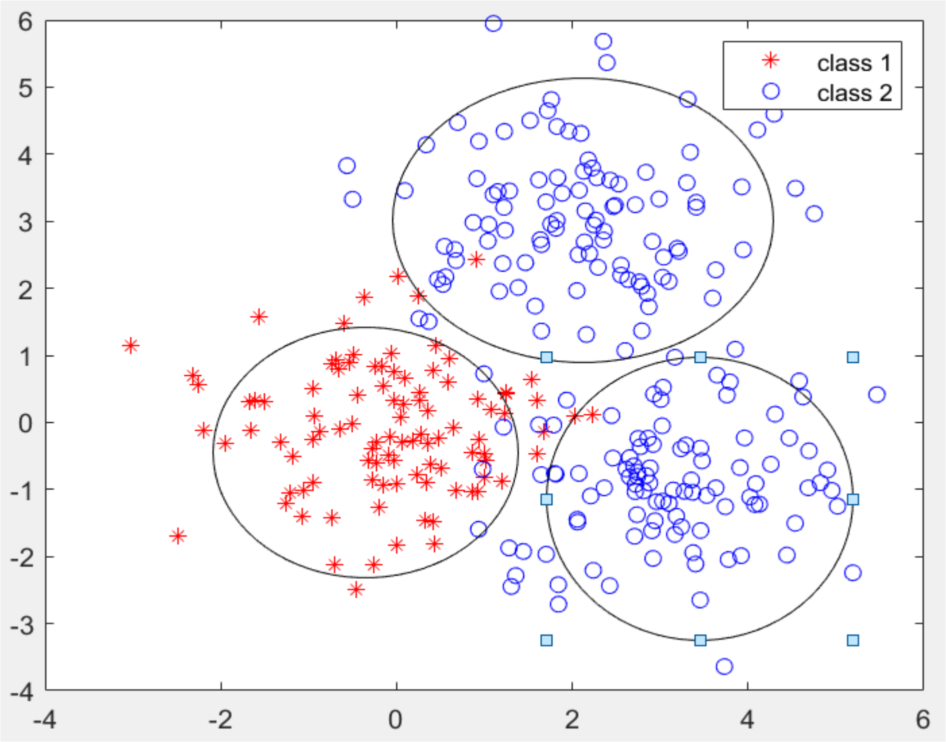
$$g_1(\mathbf{x}) = g_2(\mathbf{x})$$

Decision boundary



3.4 Gaussian Mixture Models (GMM)

A normal distribution is widely used in density estimation because the normal distribution explains data well in many cases. In some applications, however, the data distribution might be multimodal as shown below for class 2:



One way to exploit the nice properties of normal distributions for such multimodal data is to use the Gaussian Mixture Model (GMM).

A GMM model combines several normal distributions with different parameters:

$$p(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

Where m is the number of Gaussian components, α_i is the weight of the i-th component, and

$$\sum\nolimits_{i=1}^{m} \alpha_i = 1$$

3.4.1 Parameter estimation for GMM

Now imagine we know (or at least assume) the data is generated from the Gaussian mixture model. However, the parameters of the distribution remain unknown. How do we learn the parameters? As in multivariate normal distribution, we may use maximum-likelihood, or equivalently, log-likelihood) method:

$$l(\boldsymbol{\theta}) = \ln p(D|\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln p(\mathbf{x}_k|\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln \sum_{i=1}^{m} p(\mathbf{x}_k|\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

First, we define a random variable $\gamma_i(\mathbf{x}) = p(o_i|\mathbf{x})$, i.e. the probability of **x** belonging to Gaussian component o_i.

Based on Bayes theorem, we have:

$$\gamma_i(\mathbf{x}) = P(o_i|\mathbf{x}) = \frac{p(\mathbf{X}|o_i)P(o_i)}{\sum_{i=1}^{m} p(\mathbf{X}|o_i)P(o_i)} = \frac{p(\mathbf{X}|o_i)\alpha_i}{\sum_{i=1}^{m} p(\mathbf{X}|o_i)\alpha_i}$$

For the log likelihood function $l(\boldsymbol{\theta})$ to be maximum, its derivative with respect to $\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$ must be zero.

Let the derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\mu}_i$ be zero, we obtain:

$$\mu_{\boldsymbol{i}} = \frac{\sum_{k=1}^{n} \gamma_i(\mathbf{x}_k)\, \mathrm{x}_{\boldsymbol{k}}}{\sum_{k=1}^{n} \gamma_i(\mathbf{x}_k)}$$

Similarly, let the derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\Sigma}_i$ and α_i be zero, we obtain:

$$\boldsymbol{\Sigma}_{\boldsymbol{i}} = \frac{\sum_{k=1}^{n} \gamma_i(\mathbf{x}_k)(\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^T}{\sum_{k=1}^{n} \gamma_i(\mathbf{x}_k)}$$

$$\alpha_{\boldsymbol{i}} = \frac{1}{n} \sum_{k=1}^{n} \gamma_i(\mathbf{x}_k)$$

Clearly, α_i is a function of $\gamma_i(\mathbf{x}_k)$, and $\gamma_i(\mathbf{x}_k)$ is a function of α_i. Thus, the parameters cannot be estimated in closed form.

Expectation-Maximization (EM) algorithm can be used to solve the GMM parameter estimation problem.

The Expectation-Maximization (EM) algorithm is an iterative way to find maximum-likelihood estimates for model parameters

There are two basic steps in the EM algorithm, namely **E Step** or Expectation Step or Estimation Step and **M Step** or Maximization Step.

- **Estimation step:**

  Estimate $\gamma_i(\mathbf{x}_k)$ for the given values of $\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$

- **Maximization step:**

  Update $\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$ using the maximum-likelihood method

3.4.2 EM Algorithm for GMM parameter estimation

(1) Initialization

Set $j = 0$ and initialize $\hat{\alpha}_i(0), \hat{\boldsymbol{\mu}}_i(0), \hat{\boldsymbol{\Sigma}}_i(0)$ with random values

(2) Estimation

Set $j = j + 1$. Estimate $\gamma_{ik}(j)$ based on values of $\hat{\alpha}_i(j - 1), \hat{\boldsymbol{\mu}}_i(j - 1, \hat{\boldsymbol{\Sigma}}_i(j - 1)$

$$p(\mathbf{x}_{\boldsymbol{k}}|o_i) = \frac{1}{2\pi^{d/2}\sqrt{\left|\hat{\boldsymbol{\Sigma}}_i(j-1)\right|}} \exp\left[-\frac{1}{2}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i(j-1))^T \hat{\boldsymbol{\Sigma}}_i^{-1}(j-1)\,(\mathbf{x}_k - \hat{\mathbf{u}}_i(j-1))\right]$$

$$\gamma_{ik}(j) = \frac{p(\mathbf{x}_k|O_i)\hat{\alpha}_i(j-1)}{\sum_{i=1}^{m} p(\mathbf{x}_k|O_i)\hat{\alpha}_i(j-1)}$$

(3) Maximization

Update values of $\hat{\boldsymbol{\mu}}_i(j), \hat{\boldsymbol{\Sigma}}_i(j)$, $\hat{\alpha}_i(j)$ with $\gamma_{ik}(j)$:

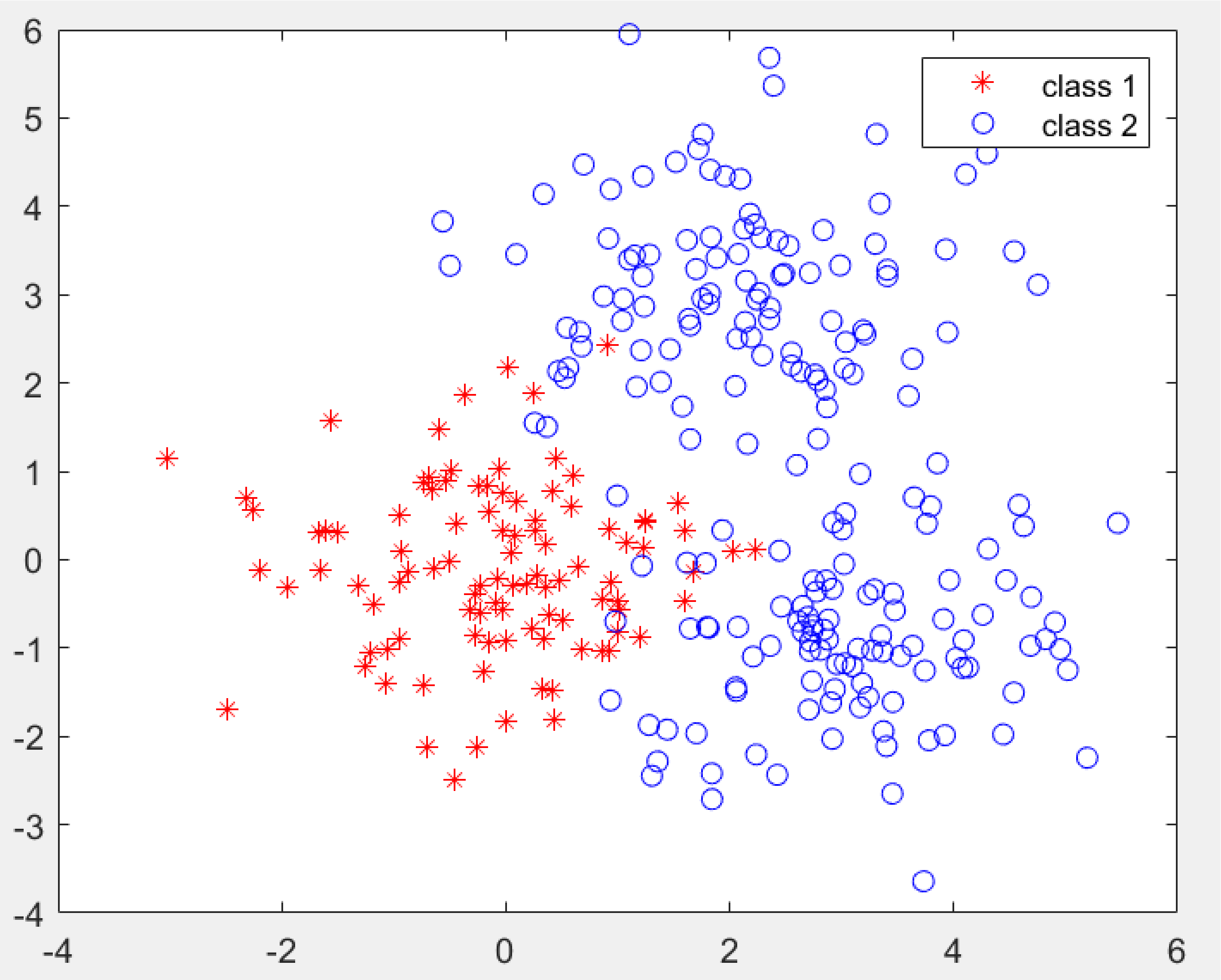
$$\hat{\mathbf{u}}_i(j) = \frac{\sum_{k=1}^{n} \gamma_{ik}(j)\, \mathrm{x}_{\boldsymbol{k}}}{\sum_{k=1}^{n} \gamma_{ik}(j)}$$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{i}}(j) = \frac{\sum_{k=1}^{n} \gamma_{ik}(j)[\mathbf{x}_k - \hat{\mathbf{u}}_i(j)][\mathbf{x}_k - \hat{\mathbf{u}}_i(j)]^T}{\sum_{k=1}^{n} \gamma_{ik}(j)}$$

$$\hat{\alpha}_i(j) = \frac{1}{n} \sum_{k=1}^{n} \gamma_{ik}(j)$$

Return to **step (2) Estimation** until the stopping criterion (such as number of iterative step is reached) is satisfied.

Example 2



To build a GMM model for samples in class 2, we use the EM algorithm.

Initialization. Set $j = 0$, assign random values to the GMM model parameters:

$$\hat{\boldsymbol{\mu}}_{21}(0) = \begin{bmatrix} -1.7729 \\ -0.0725 \end{bmatrix} \qquad \hat{\boldsymbol{\mu}}_{\mathbf{22}}(0) = \begin{bmatrix} 1.7175 \\ 0.8198 \end{bmatrix}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{21}}(\mathbf{0}) = \begin{bmatrix} -1.0055 & 1.7906 \\ -1.1064 & 2.1624 \end{bmatrix}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{22}}(\mathbf{0}) = \begin{bmatrix} -0.8193 & 1.9630 \\ -0.0370 & -0.5403 \end{bmatrix}$$

$$\hat{\alpha}_1(0) = 0.4 \qquad \hat{\alpha}_2(0) = 0.6$$

Estimation and maximization steps

After 1000 repeats of the estimation and maximization steps, we obtain the following estimates:

$$\widehat{\boldsymbol{\mu}}_{21}(1000) = \begin{bmatrix} 3.0992 \\ -0.9364 \end{bmatrix} \qquad \widehat{\boldsymbol{\mu}}_{22}(1000) = \begin{bmatrix} 2.0412 \\ 3.0483 \end{bmatrix}$$

$$\widehat{\boldsymbol{\Sigma}}_{21}(1000) = \begin{bmatrix} 1.0245 & 0.1386 \\ 0.1386 & 0.7862 \end{bmatrix}$$

$$\widehat{\boldsymbol{\Sigma}}_{22}(1000) = \begin{bmatrix} 1.1613 & -0.0057 \\ -0.0057 & 1.0778 \end{bmatrix}$$

$$\hat{\alpha}_{21}(1000) = 0.4952 \qquad \hat{\alpha}_{22}(1000) = 0.5048$$

With the estimated parameters of GMM, the class-conditional probability density function of class 2 is obtained as follows:

$$p(\mathbf{x}|\omega_2) = \sum_{i=1}^{2} \alpha_{2i} N\left(\mathbf{x}|\hat{\boldsymbol{\mu}}_{\mathbf{2}i}, \hat{\boldsymbol{\Sigma}}_{2i}\right)$$

$$= \frac{0.4952}{2\pi\sqrt{\left|\hat{\boldsymbol{\Sigma}}_{21}\right|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{21})^T \hat{\boldsymbol{\Sigma}}_{21}^{-1}\,(\mathbf{x}-\hat{\mathbf{u}}_{21})\right]$$

$$+ \frac{0.5048}{2\pi\sqrt{\left|\hat{\boldsymbol{\Sigma}}_{22}\right|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{22})^T \hat{\boldsymbol{\Sigma}}_{22}^{-1}\,(\mathbf{x}-\hat{\mathbf{u}}_{22})\right]$$

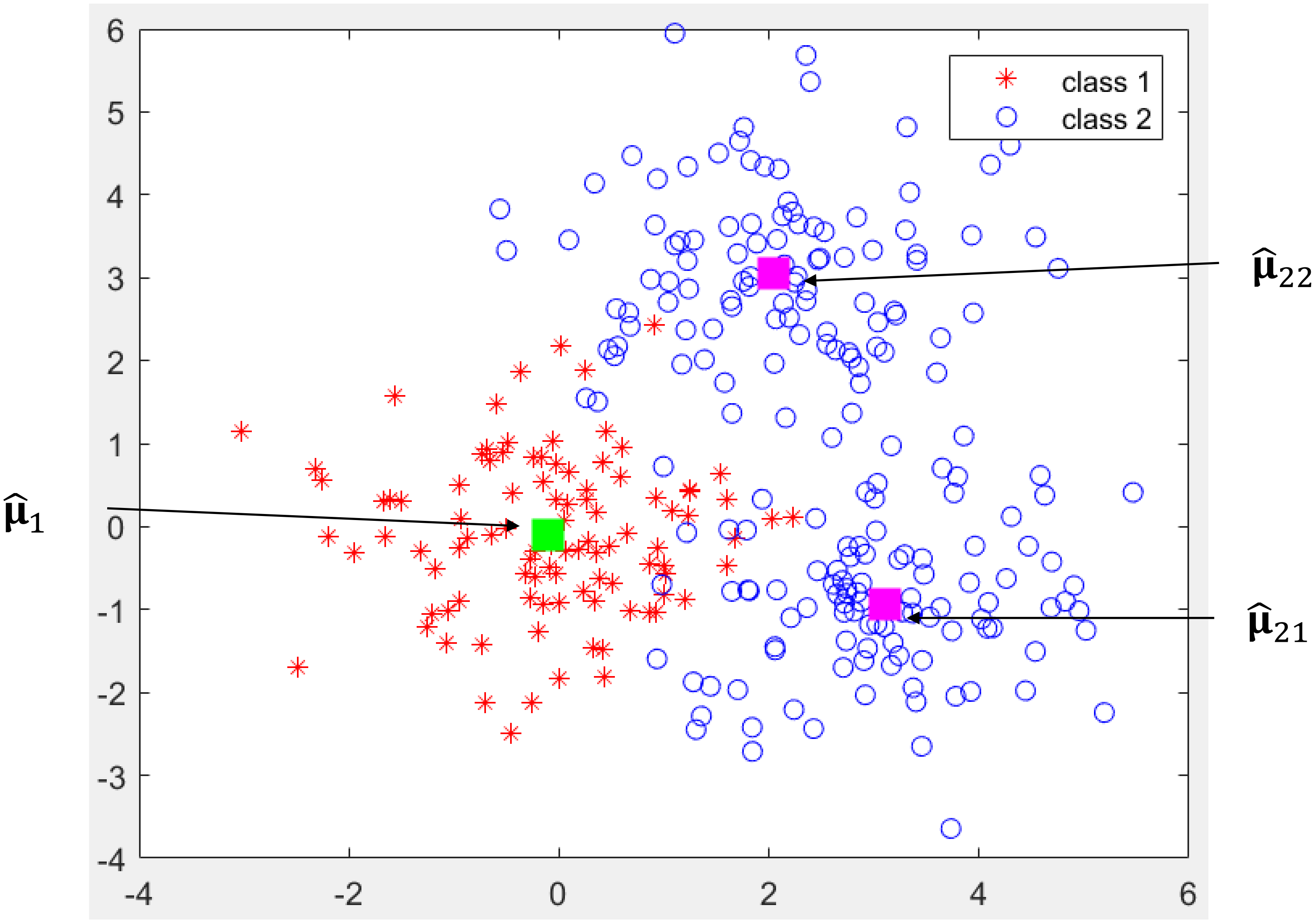
For class 1, we can use the maximum-likelihood method to estimate the mean vector $\hat{\boldsymbol{\mu}}_1$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_1$ of the single Gaussian function:

$$\hat{\boldsymbol{\mu}}_1 = \begin{bmatrix} -0.1055 \\ -0.0974 \end{bmatrix}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{1}} = \begin{bmatrix} 1.0253 & -0.0036 \\ -0.0036 & 0.8880 \end{bmatrix}$$

Then the conditional probability density function of class 1 is obtained as follows:

$$p(\mathbf{x}|\omega_1) = \frac{1}{2\pi\sqrt{|\hat{\boldsymbol{\Sigma}}_1|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_1)^T \hat{\boldsymbol{\Sigma}}_1^{-1}\,(\mathbf{x}-\hat{\mathbf{u}}_1)\right]$$


6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-2
0
2
4
6
class 1
class 2
$\hat{\boldsymbol{\mu}}_{22}$
$\hat{\boldsymbol{\mu}}_{1}$
$\hat{\boldsymbol{\mu}}_{21}$

With the class-conditional probability density functions obtained above, and the prior probabilities of the two classes :

$$p(\omega_1) = \frac{100}{300} = \frac{1}{3}$$

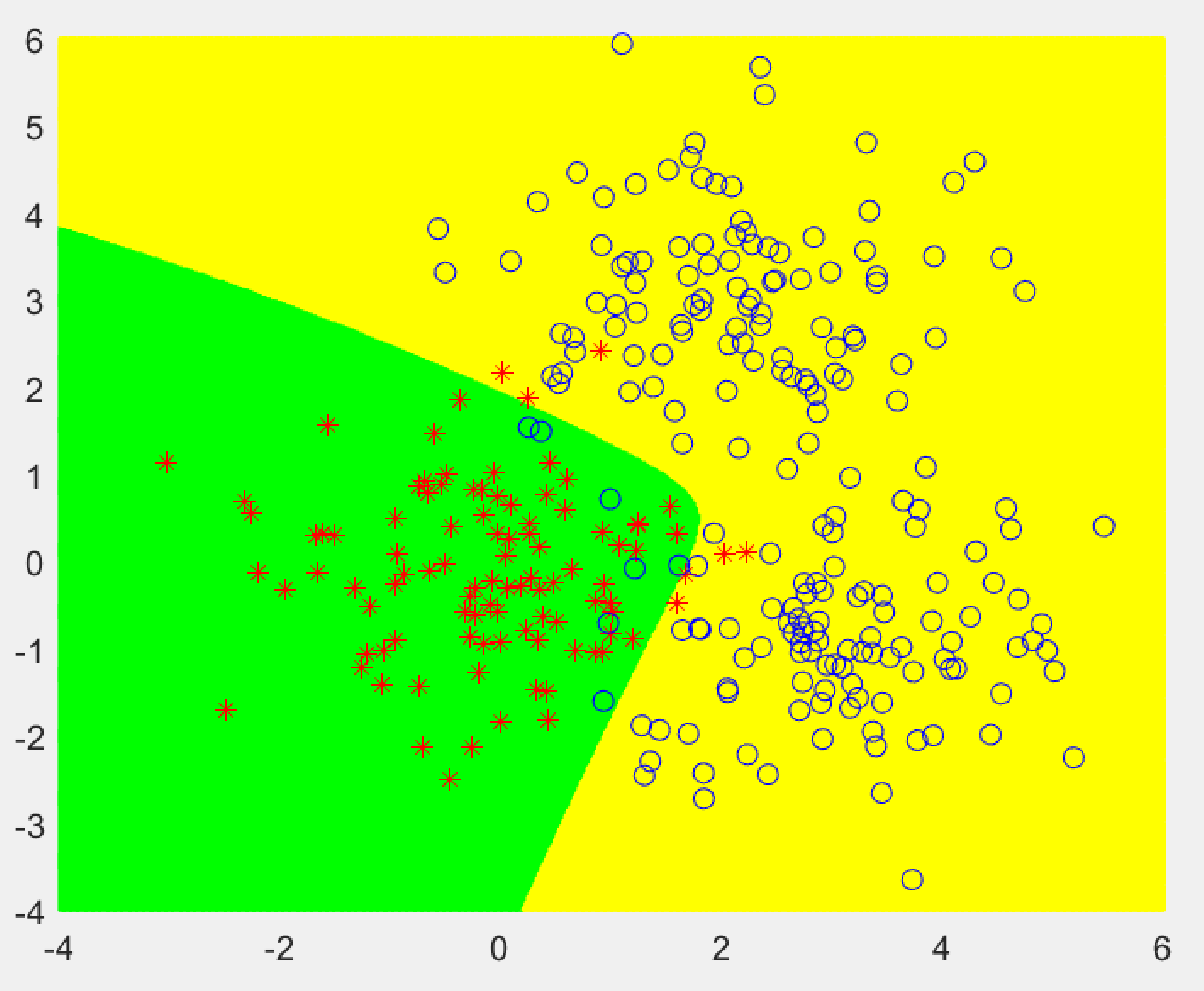
$$p(\omega_2) = \frac{200}{300} = \frac{2}{3}$$

We obtain the following discriminant functions to classify the samples in the two classes:

$$g_1(\mathbf{x}) = p(\omega_1)p(\mathbf{x}|\omega_1) = \frac{1}{3}p(\mathbf{x}|\omega_1)$$

$$g_2(\mathbf{x}) = p(\omega_2)p(\mathbf{x}|\omega_2) = \frac{2}{3}p(\mathbf{x}|\omega_2)$$

Decision boundary and 14 misclassifications



If a single Gaussian function is used to model the conditional class probability density function for class 2, by using the maximum-likelihood estimation, we have:

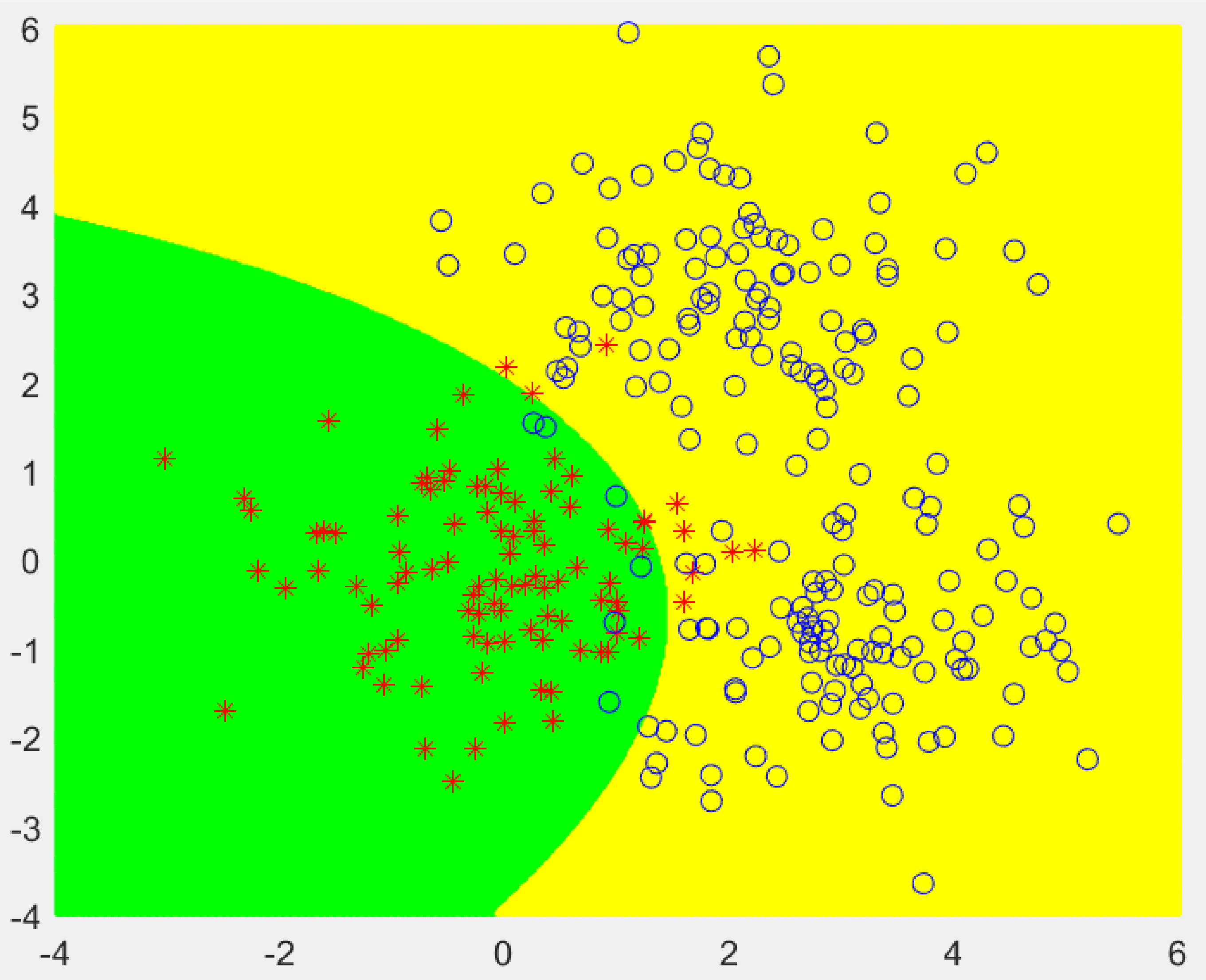
$$\hat{\boldsymbol{\mu}}_2 = \begin{bmatrix} 2.5702 \\ 1.0559 \end{bmatrix}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{2}} = \begin{bmatrix} 1.3615 & -0.9607 \\ -0.9607 & 4.8657 \end{bmatrix}$$

Then the conditional probability density function of class 2 is obtained as follows:

$$p(\mathbf{x}|\omega_2) = \frac{1}{2\pi\sqrt{|\hat{\boldsymbol{\Sigma}}_2|}} \exp\left[-\frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_2)^T \hat{\boldsymbol{\Sigma}}_2^{-1} (\mathbf{x} - \hat{\mathbf{u}}_2)\right]$$

Decision boundary and 15 misclassifications

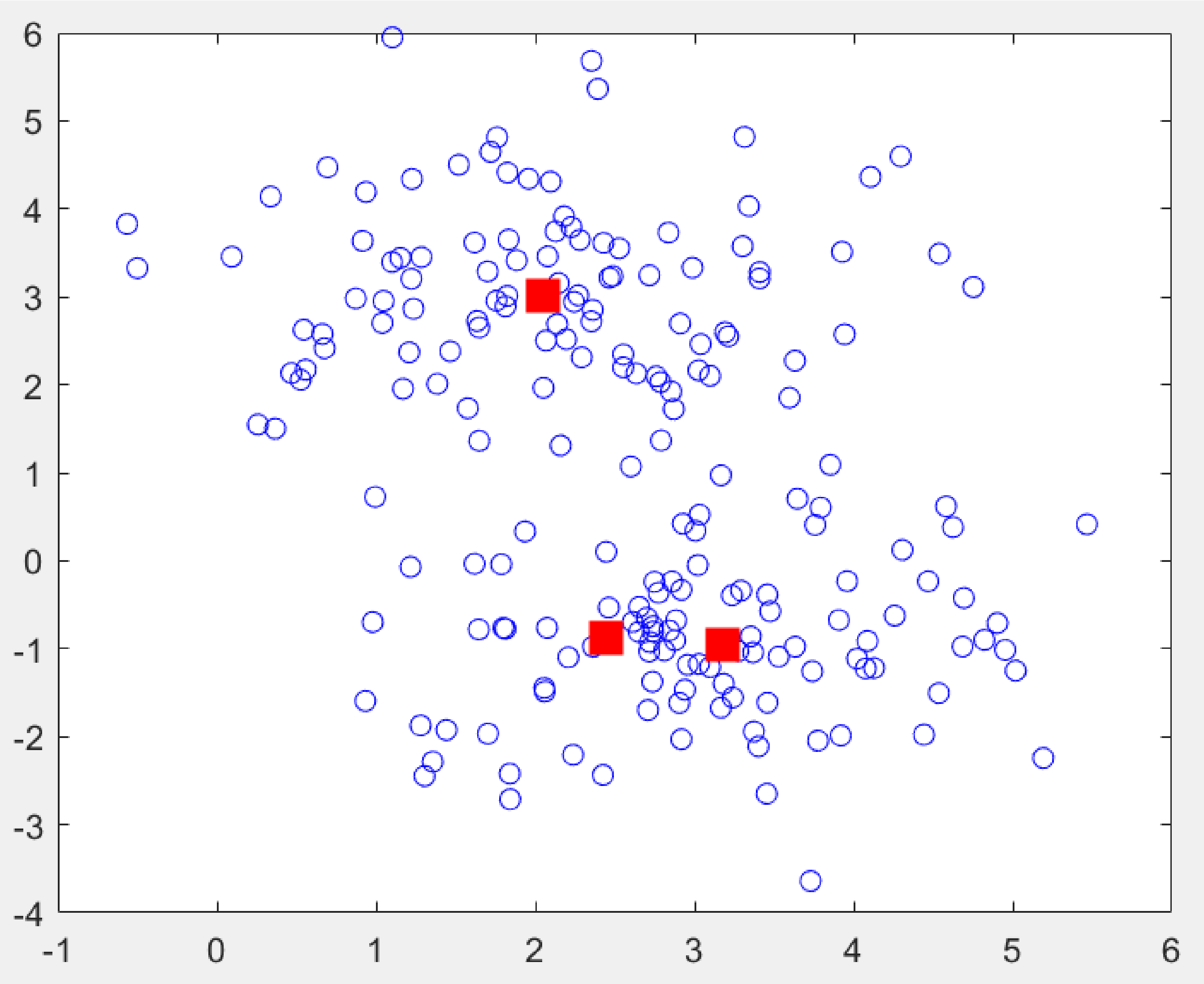


In practice, we may not know the number of Gaussian components underlying the data, in particular in high dimensional space where data is hardly to be visualized.

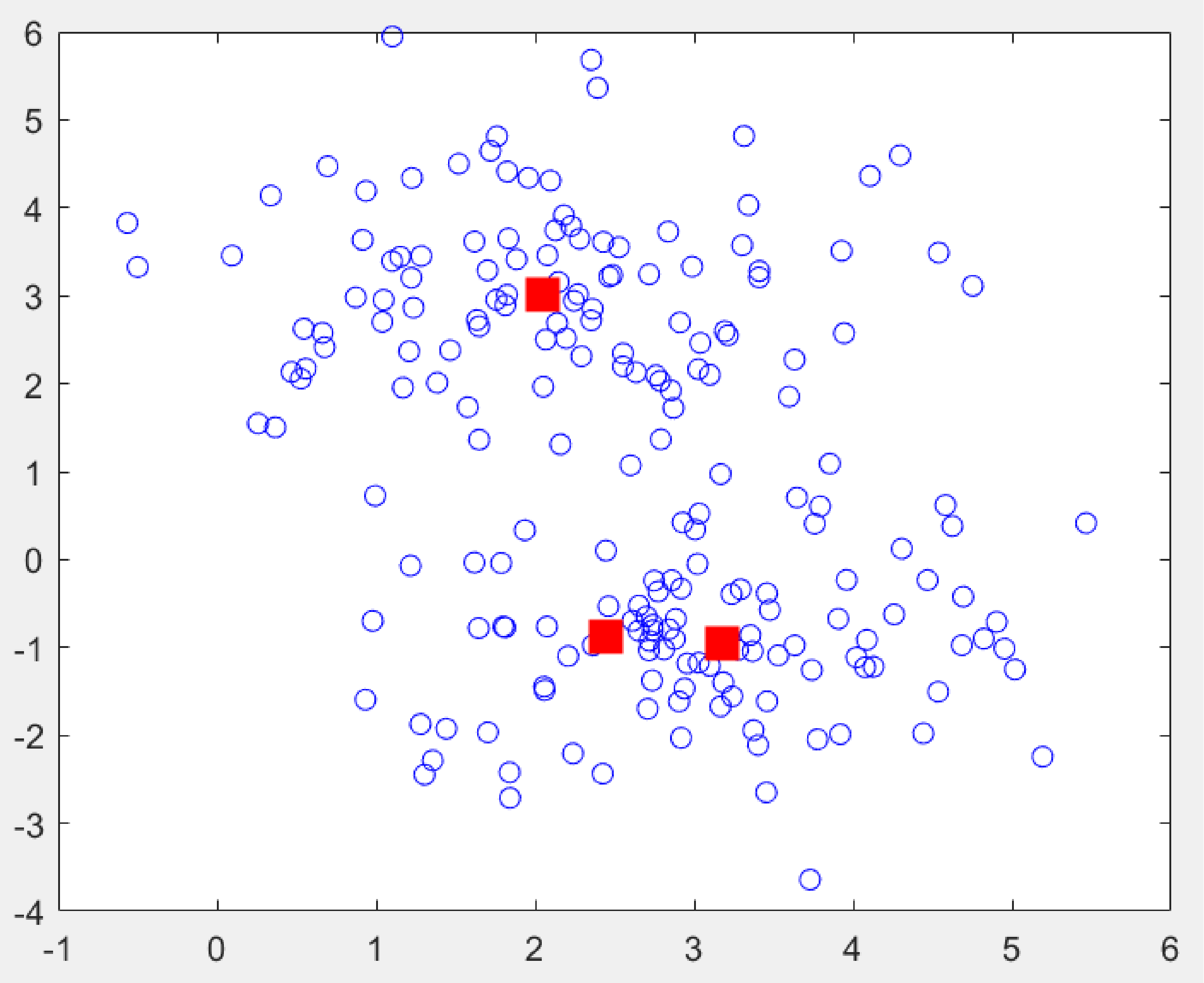
For class 2, assuming we do not know there are 2 Gaussian components. We may first guess the number of Gaussian components and then use the EM algorithm to find the parameters.

For example, we guess there are 3 Gaussian components underlying class 2:

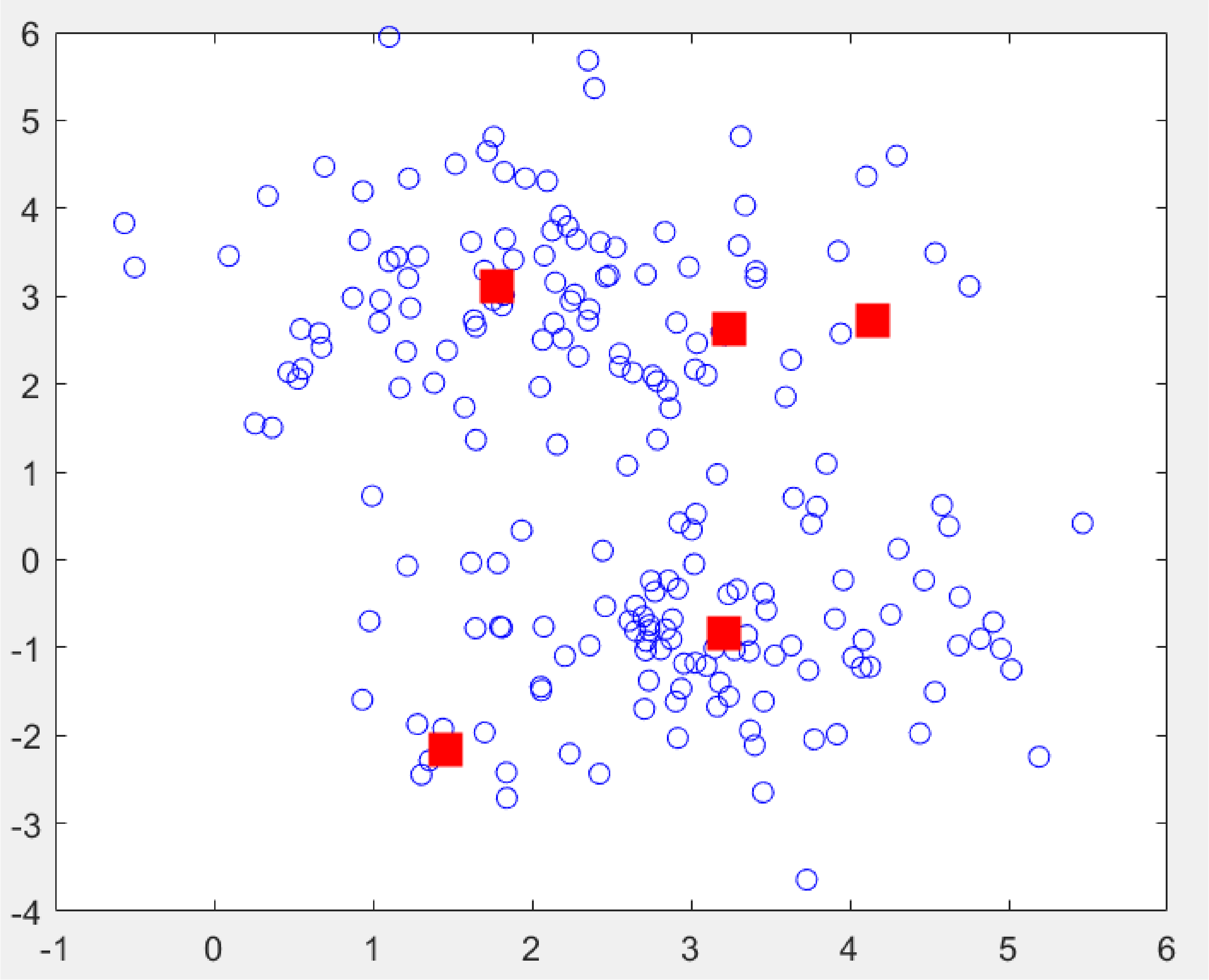
Assume there are 3 Gaussian components



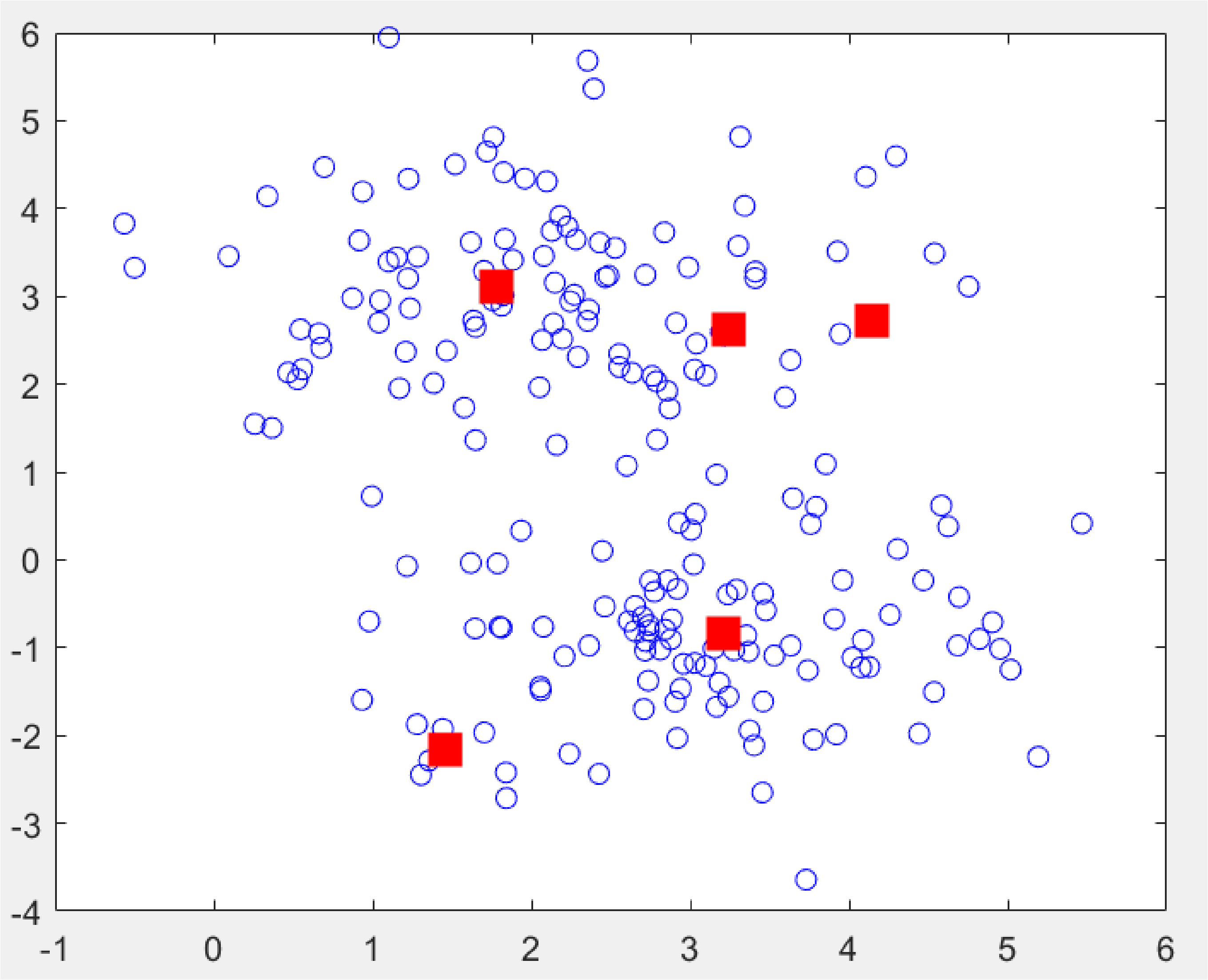
alpha=[0.0523   0.5040   0.4437]



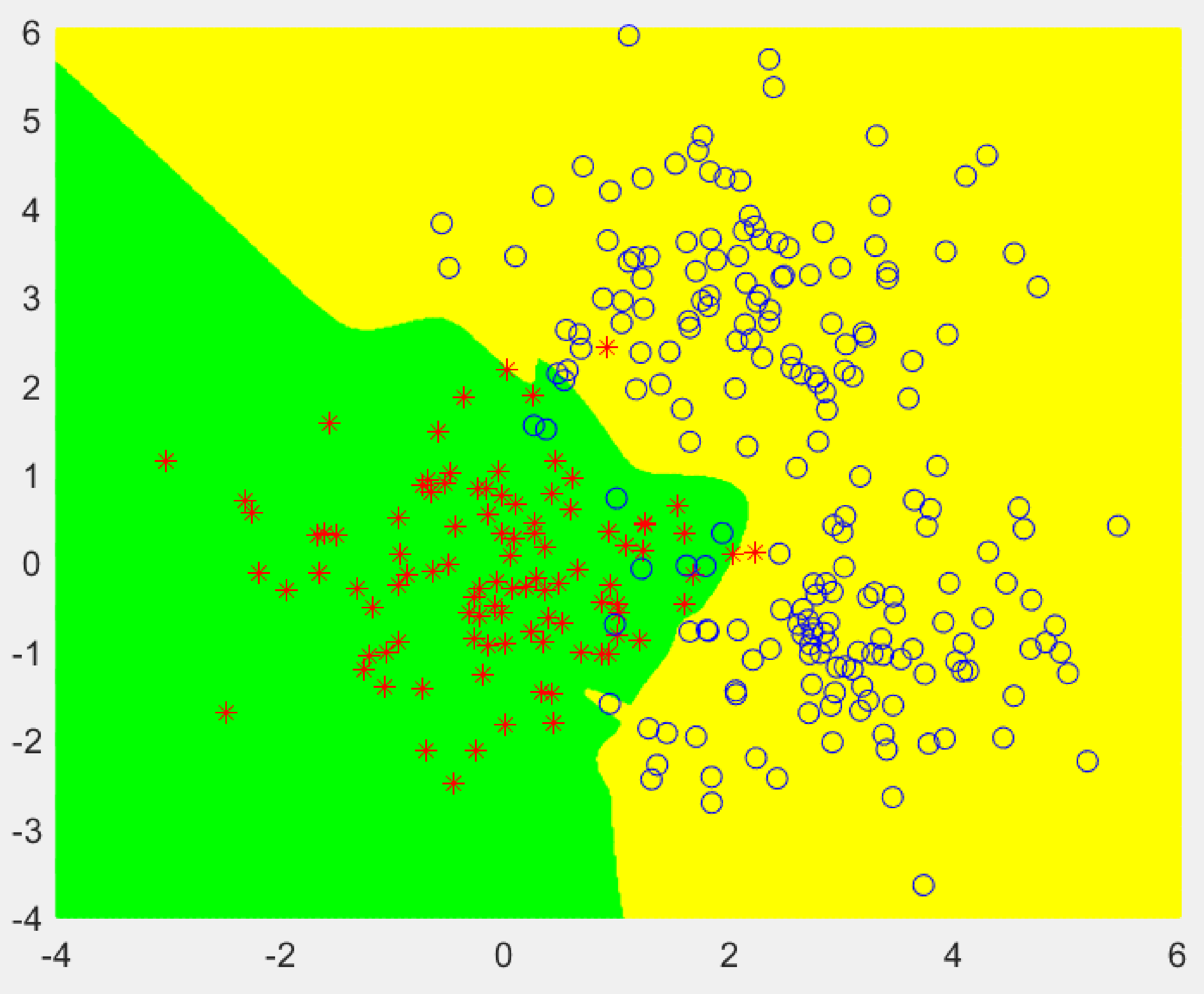
Assume there are 5 Gaussian components



Alpha=[0.4670   0.0333   0.0564   0.0206   0.4227]



Decision boundary when 2 and 5 Gaussian components are assumed for class 1 and 2, respectively



Observations:

- When excessive number of Gaussian components are used, the decision boundary (surface) becomes more complex, which may produce better performance on the training data, but worse performance on unseen testing data (i.e. overfitting problem).

- The use of excessive number of Gaussian components results in some small alpha values. This property may be used to determine suitable number of Gaussian components so as to alleviate the overfitting problem.

3.5 Naïve Bayes

3.5.1 Introduction

Based on Bayes Theorem, we have

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

Assuming feature vector $\mathbf{x}$ has n features x_1, x_2,…,x_n. The numerator is actually the joint probability of $\mathbf{x}$ and ω_j:

$$\begin{aligned}
&p(\mathbf{x}|\omega_j)p(\omega_j) = p(\mathbf{x}, \omega_j)\\
&= p(x_1, x_2, \cdots, x_n, \omega_j)\\
&= p(x_1|x_2, x_3, \cdots, x_n, \omega_j)p(x_2, x_3, \cdots, x_n, \omega_j)\\
&= p(x_1|x_2, x_3, \cdots, x_n, \omega_j)p(x_2|x_3, \cdots, x_n, \omega_j)p(x_3, \cdots, x_n, \omega_j)\\
&= p(x_1|x_2, x_3, \cdots, x_n, \omega_j)p(x_2|x_3, \cdots, x_n, \omega_j)\cdots p(x_n|\omega_j)p(\omega_j)
\end{aligned}$$

Assuming that the individual features are independent from each other. This is a strong assumption, which is not true in most practical applications and is therefore naive, hence the name. This assumption implies that

$$p(x_1 | x_2, x_3, \cdots, x_n, \omega_j) = p(x_1 | \omega_j)$$

$$\vdots$$

$$p(x_i | x_{i+1}, \cdots, x_n, \omega_j) = p(x_i | \omega_j)$$

Thus, the joint probability of $\mathbf{x}$ and ω_j is:

$$\begin{aligned} p(\mathbf{x} | \omega_j) p(\omega_j) &= p(x_1 | \omega_j) p(x_2 | \omega_j) \cdots p(x_n | \omega_j) p(\omega_j) \\ &= \prod_{i=1}^{n} p(x_i | \omega_j) p(\omega_j) \end{aligned}$$

Then the posterior probability is:

$$p(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)p(\omega_j)}{p(\mathbf{x})} = \frac{\prod_{i=1}^{n} p(x_i|\omega_j)p(\omega_j)}{p(\mathbf{x})}$$

3.5.2 Types of Naïve Bayes Classifier:

There are three types of Naive Bayes Classifiers:

(1) Gaussian Naive Bayes - used for continuous data.

(2) Bernoulli Naive Bayes: used for discrete data whose features have binary values.

(3) Multinomial Naive Bayes: often used for text classification where the count of words in the text is used to represent the text.

3.5.2.1 Gaussian Naive Bayes

Gaussian Naive Bayes is used for continuous data. For each continuous feature x, assuming it follows normal distribution, then the class conditional probability density function is as follow:

$$p(x|\omega_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_j}{\sigma_j}\right)^2\right]$$

Where μ_j and σ_j denote the mean and standard deviation of feature x for class ω_j, which can be easily estimated from samples of class ω_j using the maximum likelihood method.

3.5.2.2 Bernoulli Naive Bayes

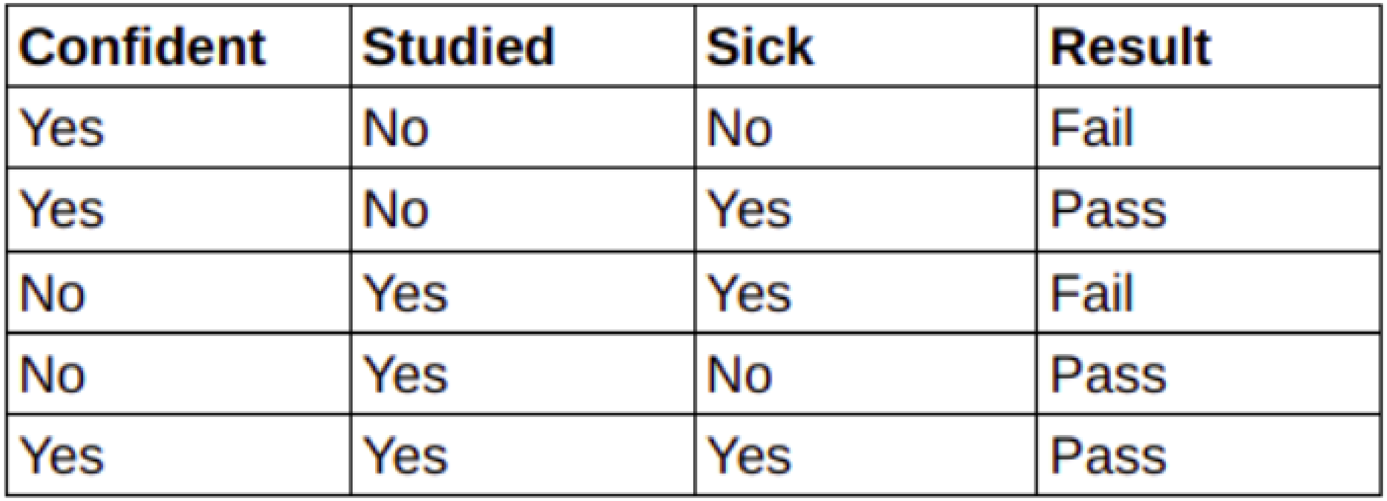
Bernoulli Naive Bayes is used for discrete data and it works on Bernoulli distribution. The main feature of Bernoulli Naive Bayes is that it assumes binary values like true or false, yes or no, success or failure, 0 or 1 and so on.

Bernoulli distribution

As we deal with binary values, let's consider r as the probability of success and q as probability of failure, then $q = 1 - r$. For a random variable X in Bernoulli distribution,

$$p(x) = P(X = x) = \begin{cases} r, & x = 1 \\ 1 - r, & x = 0 \end{cases}$$

Consider the following dataset showing the result whether a person Pass (ω_1) or Fail (ω_2) in the exam:



Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

Our task is to classify instance **x** with Confident=Yes, Studied=Yes and Sick=No.

Assuming that the result of Pass and Fail are denoted by ω_1 and ω_2, respectively. Features Confident, Studied and Sick are denoted by X_1, X_2 and X_3 respectively. Then:

$$p\left(\omega_j \middle| \mathbf{x}\right) = \frac{p\left(\mathbf{x} \middle| \omega_j\right) p(\omega_j)}{p(\mathbf{x})} = \frac{\prod_{i=1}^{3} p\left(x_i \middle| \omega_j\right) p(\omega_j)}{p(\mathbf{x})}$$

For a sample with $x_1 = Yes$, $x_2 = Yes$, $x_3 = No$, to obtain the posterior probabilities, we need to first find:

(1) prior probabilities $p\left(\omega_j\right), j = 1{,}2$, and

(2) class conditional probabilities

$$p(x_1 = Yes|\omega_j)$$

$$p(x_2 = Yes|\omega_j)$$

$$p\left(x_3 = No \middle| \omega_j\right)$$

First, we calculate the prior probabilities:

$$p(\omega_1) = \frac{3}{5} = 0.6$$

$$p(\omega_2) = \frac{2}{5} = 0.4$$

Second, we calculate the class conditional probability for each individual feature.

$$p(x_1 = Yes|\omega_1) = \frac{2}{3}$$

$$p(x_2 = Yes|\omega_1) = \frac{2}{3}$$

$$p(x_3 = No|\omega_1) = \frac{1}{3}$$

Similarly,

$$p(x_1 = Yes|\omega_2) = \frac{1}{2}$$

$$p(x_2 = Yes|\omega_2) = \frac{1}{2}$$

$$p(x_3 = No|\omega_2) = \frac{1}{2}$$

Hence:

$$p(\mathbf{x}|\omega_1)p(\omega_1) = \frac{2}{3}\times\frac{2}{3}\times\frac{1}{3}\times\frac{3}{5} = \frac{4}{45} = 0.088$$

$$p(\mathbf{x}|\omega_2)p(\omega_2) = \frac{1}{2}\times\frac{1}{2}\times\frac{1}{2}\times\frac{2}{5} = \frac{1}{20} = 0.05$$

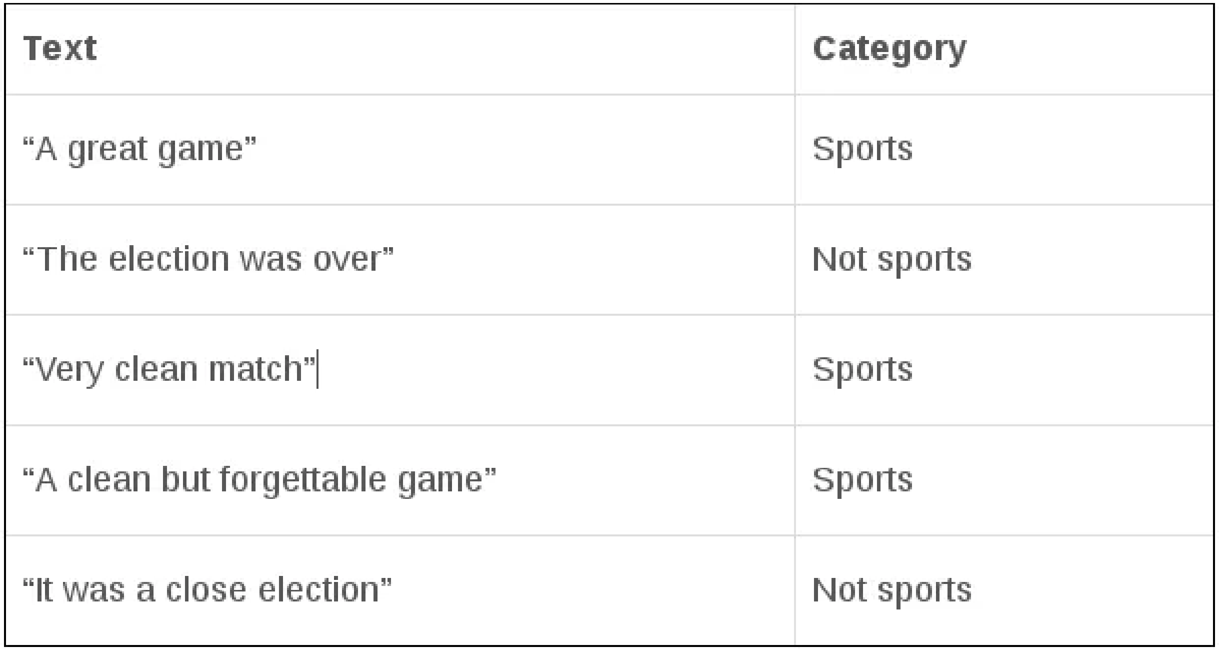
$p(\mathbf{x})$ is a common denominator, we can ignore it. Since

$$p(\mathbf{x}|\omega_1)p(\omega_1) > p(\mathbf{x}|\omega_2)p(\omega_2)$$

The instance with 'Confident=Yes, Studied=Yes and Sick=No’ can be predicted as 'Pass'

3.5.2.3 Multinomial Naïve Bayes

Multinominal Naïve Bayes is widely used for document (text) classification. So we explain it in the context of text classification. Consider the following problem with 5 labelled training samples:



Text	Category
“A great game”	Sports
“The election was over”	Not sports
“Very clean match”	Sports
“A clean but forgettable game”	Sports
“It was a close election”	Not sports

Our task is to classify the text “A very close game” into one of the two categories: Sports and Not sports.

$$p(Sports|A\ very\ close\ game)$$

$$=\frac{p(A\ very\ close\ game|Sports)p(Sports)}{p(A\ very\ close\ game)}$$

$$=\frac{p(A|Sports)p(very|Sports)\cdots p(game|Sports)p(Sports)}{p(A\ very\ close\ game)}$$

Similarly,

$$p(Not\ sports|A\ very\ close\ game)$$

$$=\frac{p(A\ very\ close\ game|Not\ sports)p(Not\ sports)}{p(A\ very\ close\ game)}$$

$$=\frac{p(A|Not\ sports)\cdots p(game|Not\ sports)p(Not\ sports)}{p(A\ very\ close\ game)}$$

Assuming the i^{th} word is denoted by x_i, and the categories Sports and Not sports are denoted by ω_1 and ω_2, respectively. The class conditional probability $p(x_i|\omega_j)$ can be estimated from training data using the following formula:

$$p(x_i|\omega_j) = \frac{counts(x_i, \omega_j)}{\sum_{k=1}^{d} counts(\, x_k, \omega_j)}$$

Where $counts(x_k, \omega_j)$ denotes the total number of occurrence of word x_k in the training data belonging to category ω_j. d is the number of unique words in all training data.

But there is one issue in the above estimation. If a word is not seen in the training data, then the conditional probability will be zero, leading to zero posterior probability.

For example, for word “close”,

$$p(close|Sports) = \frac{0}{11} = 0$$

To address the issue, we can use the so-called Laplace smoothing (add 1):

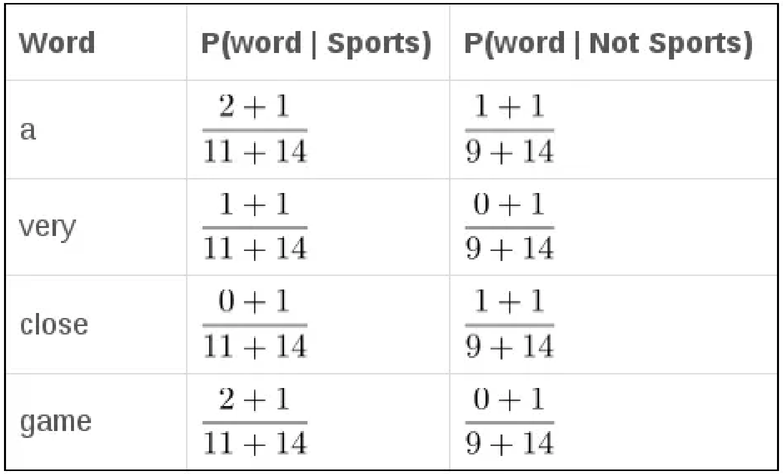
$$p(x_i|\omega_j) = \frac{counts(x_i, \omega_j) + 1}{\sum_{k=1}^{d}(counts(\, x_k, \omega_j) + 1)}$$

$$= \frac{counts(x_i,\omega_j)+1}{(\sum_{k=1}^{d} counts(x_k,\omega_j))+d}$$

After Laplace smoothing,

$$p(close|Sports) = \frac{0+1}{11+14} = \frac{1}{25}$$

Accordingly, we can obtain the other class conditional probabilities:



Word	P(word | Sports)	P(word | Not Sports)
a	$\frac{2+1}{11+14}$	$\frac{1+1}{9+14}$
very	$\frac{1+1}{11+14}$	$\frac{0+1}{9+14}$
close	$\frac{0+1}{11+14}$	$\frac{1+1}{9+14}$
game	$\frac{2+1}{11+14}$	$\frac{0+1}{9+14}$

$$p(Sports|A\ very\ close\ game)$$

$$= \frac{p(A|Sports)p(very|Sports)\cdots p(game|Sports)p(Sports)}{p(A\ very\ close\ game)}$$

$$= \frac{\frac{3}{25}\times\frac{2}{25}\times\frac{1}{25}\times\frac{3}{25}}{p(A\ very\ close\ game)} = \frac{4.61\times 10^{-5}}{p(A\ very\ close\ game)}$$

$$p(Not\ sports|A\ very\ close\ game)$$

$$= \frac{p(A|Not\ sports) \cdots p(game|Not\ sports)p(Not\ sports)}{p(A\ very\ close\ game)}$$

$$= \frac{\frac{2}{23} \times \frac{1}{23} \times \frac{2}{23} \times \frac{1}{23}}{p(A\ very\ close\ game)} = \frac{1.43 \times 10^{-5}}{p(A\ very\ close\ game)}$$

Because

$$p(Sports|A\ very\ close\ game) > p(Not\ sports|A\ very\ close\ game)$$

The text “ A very close game” is classified as Sports.

Notes:

(1) In spite of the apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations such as document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters.

(2) Naive Bayes classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.