

# Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki<sup>1</sup>    Wagner Meira Jr.<sup>2</sup>

<sup>1</sup>Department of Computer Science  
Rensselaer Polytechnic Institute, Troy, NY, USA

<sup>2</sup>Department of Computer Science  
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 18: Probabilistic Classification

# Bayes Classifier

Let the training dataset  $\mathbf{D}$  consists of  $n$  points  $\mathbf{x}_i$  in a  $d$ -dimensional space, and let  $y_i$  denote the class for each point, with  $y_i \in \{c_1, c_2, \dots, c_k\}$ .

The Bayes classifier estimates the posterior probability  $P(c_i|\mathbf{x})$  for each class  $c_i$ , and chooses the class that has the largest probability. The predicted class for  $\mathbf{x}$  is given as

$$\hat{y} = \arg \max_{c_i} \{P(c_i|\mathbf{x})\}$$

According to the Bayes theorem, we have

$$P(c_i|\mathbf{x}) = \frac{P(\mathbf{x}|c_i) \cdot P(c_i)}{P(\mathbf{x})}$$

Because  $P(\mathbf{x})$  is fixed for a given point, Bayes rule can be rewritten as

$$\hat{y} = \arg \max_{c_i} \{P(c_i|\mathbf{x})\} = \arg \max_{c_i} \left\{ \frac{P(\mathbf{x}|c_i)P(c_i)}{P(\mathbf{x})} \right\} = \arg \max_{c_i} \{P(\mathbf{x}|c_i)P(c_i)\}$$

## Estimating the Prior Probability: $P(c_i)$

Let  $D_i$  denote the subset of points in  $D$  that are labeled with class  $c_i$ :

$$D_i = \{\mathbf{x}_j \in D \mid \mathbf{x}_j \text{ has class } y_j = c_i\}$$

Let the size of the dataset  $D$  be given as  $|D| = n$ , and let the size of each class-specific subset  $D_i$  be given as  $|D_i| = n_i$ .

The prior probability for class  $c_i$  can be estimated as follows:

$$\hat{P}(c_i) = \frac{n_i}{n}$$

# Estimating the Likelihood

## Numeric Attributes, Parametric Approach

To estimate the likelihood  $P(\mathbf{x}|c_i)$ , we have to estimate the joint probability of  $\mathbf{x}$  across all the  $d$  dimensions, i.e., we have to estimate  $P(\mathbf{x} = (x_1, x_2, \dots, x_d)|c_i)$ .

In the parametric approach we assume that each class  $c_i$  is normally distributed, and we use the estimated mean  $\hat{\boldsymbol{\mu}}_i$  and covariance matrix  $\hat{\boldsymbol{\Sigma}}_i$  to compute the probability density at  $\mathbf{x}$

$$\hat{f}_i(\mathbf{x}) = \hat{f}(\mathbf{x}|\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i) = \frac{1}{(\sqrt{2\pi})^d \sqrt{|\hat{\boldsymbol{\Sigma}}_i|}} \exp \left\{ -\frac{(\mathbf{x} - \hat{\boldsymbol{\mu}}_i)^T \hat{\boldsymbol{\Sigma}}_i^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_i)}{2} \right\}$$

The posterior probability is then given as

$$P(c_i|\mathbf{x}) = \frac{\hat{f}_i(\mathbf{x})P(c_i)}{\sum_{j=1}^k \hat{f}_j(\mathbf{x})P(c_j)}$$

The predicted class for  $\mathbf{x}$  is:

$$\hat{y} = \arg \max_{c_i} \left\{ \hat{f}_i(\mathbf{x})P(c_i) \right\}$$

# Bayes Classifier Algorithm

**BayesClassifier** ( $D = \{(\mathbf{x}_j, y_j)\}_{j=1}^n$ ):

```
1 for  $i = 1, \dots, k$  do
2    $D_i \leftarrow \{\mathbf{x}_j \mid y_j = c_i, j = 1, \dots, n\}$  // class-specific subsets
3    $n_i \leftarrow |D_i|$  // cardinality
4    $\hat{P}(c_i) \leftarrow n_i/n$  // prior probability
5    $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\mathbf{x}_j \in D_i} \mathbf{x}_j$  // mean
6    $\mathbf{Z}_i \leftarrow D_i - \mathbf{1}_{n_i} \hat{\boldsymbol{\mu}}_i^T$  // centered data
7    $\hat{\boldsymbol{\Sigma}}_i \leftarrow \frac{1}{n_i} \mathbf{Z}_i^T \mathbf{Z}_i$  // covariance matrix
8 return  $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i$  for all  $i = 1, \dots, k$ 
```

**Testing** ( $\mathbf{x}$  and  $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i$ , for all  $i \in [1, k]$ ):

```
9  $\hat{y} \leftarrow \arg \max_{c_i} \{f(\mathbf{x} | \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i) \cdot P(c_i)\}$ 
10 return  $\hat{y}$ 
```

## Bayes Classifier: Iris Data

Consider the 2D Iris data (sepal length and sepal width). Class  $c_1$ , which corresponds to *iris-setosa* (circles), has  $n_1 = 50$  points, whereas the other class  $c_2$  (triangles) has  $n_2 = 100$  points.

$$\hat{P}(c_1) = \frac{n_1}{n} = \frac{50}{150} = 0.33$$

$$\hat{P}(c_2) = \frac{n_2}{n} = \frac{100}{150} = 0.67$$

$$\hat{\mu}_1 = \begin{pmatrix} 5.006 \\ 3.418 \end{pmatrix}$$

$$\hat{\mu}_2 = \begin{pmatrix} 6.262 \\ 2.872 \end{pmatrix}$$

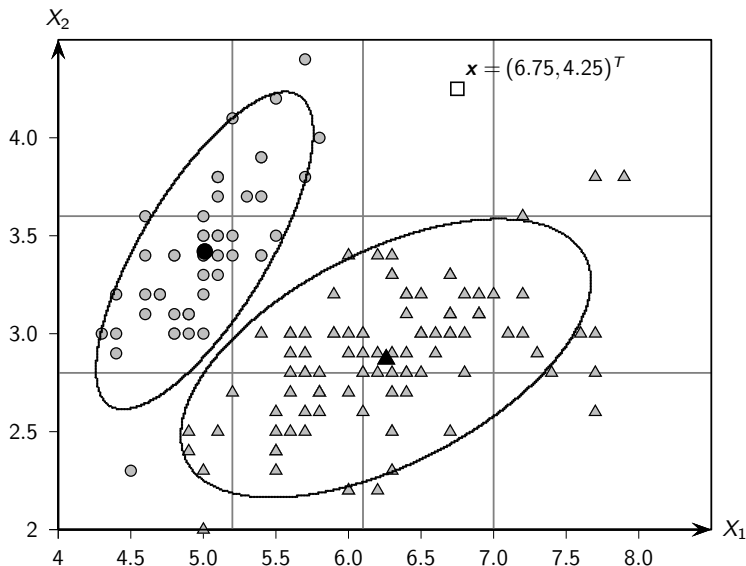
$$\hat{\Sigma}_1 = \begin{pmatrix} 0.1218 & 0.0983 \\ 0.0983 & 0.1423 \end{pmatrix}$$

$$\hat{\Sigma}_2 = \begin{pmatrix} 0.435 & 0.1209 \\ 0.1209 & 0.1096 \end{pmatrix}$$

Plot shows the contour or level curve (corresponding to 1% of the peak density) for the multivariate normal distribution modeling the probability density for both classes.

# Bayes Classifier: Iris Data

$X_1$ : sepal length versus  $X_2$ : sepal width



Let  $\mathbf{x} = (6.75, 4.25)^T$  be a test point (shown as white square). The posterior probabilities for  $c_1$  and  $c_2$  can be computed as:

$$\hat{P}(c_1|\mathbf{x}) \propto \hat{f}(\mathbf{x}|\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1)\hat{P}(c_1) = (4.951 \times 10^{-7}) \times 0.33 = 1.634 \times 10^{-7}$$

$$\hat{P}(c_2|\mathbf{x}) \propto \hat{f}(\mathbf{x}|\hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}_2)\hat{P}(c_2) = (2.589 \times 10^{-5}) \times 0.67 = 1.735 \times 10^{-5}$$

Because  $\hat{P}(c_2|\mathbf{x}) > \hat{P}(c_1|\mathbf{x})$  the class for  $\mathbf{x}$  is predicted as  $\hat{y} = c_2$ .



# Bayes Classifier

## Categorical Attributes

Let  $X_j$  be a categorical attribute over the domain  $dom(X_j) = \{a_{j1}, a_{j2}, \dots, a_{jm_j}\}$ . Each categorical attribute  $X_j$  is modeled as an  $m_j$ -dimensional multivariate Bernoulli random variable  $\mathbf{X}_j$  that takes on  $m_j$  distinct vector values  $\mathbf{e}_{j1}, \mathbf{e}_{j2}, \dots, \mathbf{e}_{jm_j}$ , where  $\mathbf{e}_{jr}$  is the  $r$ th standard basis vector in  $\mathbb{R}^{m_j}$  and corresponds to the  $r$ th value or symbol  $a_{jr} \in dom(X_j)$ .

The entire  $d$ -dimensional dataset is modeled as the vector random variable  $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_d)^T$ . Let  $d' = \sum_{j=1}^d m_j$ ; a categorical point  $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$  is therefore represented as the  $d'$ -dimensional binary vector

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_d \end{pmatrix} = \begin{pmatrix} \mathbf{e}_{1r_1} \\ \vdots \\ \mathbf{e}_{dr_d} \end{pmatrix}$$

where  $\mathbf{v}_j = \mathbf{e}_{jr_j}$  provided  $x_j = a_{jr_j}$  is the  $r_j$ th value in the domain of  $X_j$ .

# Bayes Classifier

## Categorical Attributes

The probability of the categorical point  $\mathbf{x}$  is obtained from the joint probability mass function (PMF) for the vector random variable  $\mathbf{X}$ :

$$P(\mathbf{x}|c_i) = f(\mathbf{v}|c_i) = f(\mathbf{X}_1 = \mathbf{e}_{1r_1}, \dots, \mathbf{X}_d = \mathbf{e}_{dr_d} | c_i)$$

The joint PMF can be estimated directly from the data  $\mathbf{D}_i$  for each class  $c_i$  as follows:

$$\hat{f}(\mathbf{v}|c_i) = \frac{n_i(\mathbf{v})}{n_i}$$

where  $n_i(\mathbf{v})$  is the number of times the value  $\mathbf{v}$  occurs in class  $c_i$ .

However, to avoid zero probabilities we add a *pseudo-count* of 1 for each value

$$\hat{f}(\mathbf{v}|c_i) = \frac{n_i(\mathbf{v}) + 1}{n_i + \prod_{j=1}^d m_j}$$

# Discretized Iris Data

Assume that the sepal length and sepal width attributes in the Iris dataset have been discretized as shown below.

Bins	Domain
[4.3, 5.2]	Very Short ( $a_{11}$ )
(5.2, 6.1]	Short ( $a_{12}$ )
(6.1, 7.0]	Long ( $a_{13}$ )
(7.0, 7.9]	Very Long ( $a_{14}$ )

(a) Discretized sepal length

Bins	Domain
[2.0, 2.8]	Short ( $a_{21}$ )
(2.8, 3.6]	Medium ( $a_{22}$ )
(3.6, 4.4]	Long ( $a_{23}$ )

(b) Discretized sepal width

We have  $|dom(X_1)| = m_1 = 4$  and  $|dom(X_2)| = m_2 = 3$ .

# Discretized Iris Data

## Class-specific Empirical Joint Probability Mass Function

Class: $c_1$		$X_2$			$\hat{f}_{X_1}$
		Short ( $e_{21}$ )	Medium ( $e_{22}$ )	Long ( $e_{23}$ )	
$X_1$	Very Short ( $e_{11}$ )	1/50	33/50	5/50	39/50
	Short ( $e_{12}$ )	0	3/50	8/50	13/50
	Long ( $e_{13}$ )	0	0	0	0
	Very Long ( $e_{14}$ )	0	0	0	0
$\hat{f}_{X_2}$		1/50	36/50	13/50	

Class: $c_2$		$X_2$			$\hat{f}_{X_1}$
		Short ( $e_{21}$ )	Medium ( $e_{22}$ )	Long ( $e_{23}$ )	
$X_1$	Very Short ( $e_{11}$ )	6/100	0	0	6/100
	Short ( $e_{12}$ )	24/100	15/100	0	39/100
	Long ( $e_{13}$ )	13/100	30/100	0	43/100
	Very Long ( $e_{14}$ )	3/100	7/100	2/100	12/100
$\hat{f}_{X_2}$		46/100	52/100	2/100	

## Discretized Iris Data: Test Case

Consider a test point  $\mathbf{x} = (5.3, 3.0)^T$  corresponding to the categorical point (Short, Medium), which is represented as  $\mathbf{v} = (\mathbf{e}_{12}^T \quad \mathbf{e}_{22}^T)^T$ .

The prior probabilities of the classes are  $\hat{P}(c_1) = 0.33$  and  $\hat{P}(c_2) = 0.67$ . The likelihood and posterior probability for each class is given as

$$\hat{P}(\mathbf{x}|c_1) = \hat{f}(\mathbf{v}|c_1) = 3/50 = 0.06$$

$$\hat{P}(\mathbf{x}|c_2) = \hat{f}(\mathbf{v}|c_2) = 15/100 = 0.15$$

$$\hat{P}(c_1|\mathbf{x}) \propto 0.06 \times 0.33 = 0.0198$$

$$\hat{P}(c_2|\mathbf{x}) \propto 0.15 \times 0.67 = 0.1005$$

In this case the predicted class is  $\hat{y} = c_2$ .

## Discretized Iris Data: Test Case with Pseudo-counts

The test point  $\mathbf{x} = (6.75, 4.25)^T$  corresponds to the categorical point (Long, Long), and it is represented as  $\mathbf{v} = (\mathbf{e}_{13}^T \quad \mathbf{e}_{23}^T)^T$ .

Unfortunately the probability mass at  $\mathbf{v}$  is zero for both classes. We adjust the PMF via pseudo-counts noting that the number of possible values are  $m_1 \times m_2 = 4 \times 3 = 12$ .

The likelihood and prior probability can then be computed as

$$\hat{P}(\mathbf{x}|c_1) = \hat{f}(\mathbf{v}|c_1) = \frac{0+1}{50+12} = 1.61 \times 10^{-2}$$

$$\hat{P}(\mathbf{x}|c_2) = \hat{f}(\mathbf{v}|c_2) = \frac{0+1}{100+12} = 8.93 \times 10^{-3}$$

$$\hat{P}(c_1|\mathbf{x}) \propto (1.61 \times 10^{-2}) \times 0.33 = 5.32 \times 10^{-3}$$

$$\hat{P}(c_2|\mathbf{x}) \propto (8.93 \times 10^{-3}) \times 0.67 = 5.98 \times 10^{-3}$$

Thus, the predicted class is  $\hat{y} = c_2$ .

# Bayes Classifier: Challenges

The main problem with the Bayes classifier is the lack of enough data to reliably estimate the joint probability density or mass function, especially for high-dimensional data.

For numeric attributes we have to estimate  $O(d^2)$  covariances, and as the dimensionality increases, this requires us to estimate too many parameters.

For categorical attributes we have to estimate the joint probability for all the possible values of  $\mathbf{v}$ , given as  $\prod_j |\text{dom}(X_j)|$ . Even if each categorical attribute has only two values, we would need to estimate the probability for  $2^d$  values. However, because there can be at most  $n$  distinct values for  $\mathbf{v}$ , most of the counts will be zero.

Naive Bayes classifier addresses these concerns.

# Naive Bayes Classifier

## Numeric Attributes

The naive Bayes approach makes the simple assumption that all the attributes are independent, which implies that the likelihood can be decomposed into a product of dimension-wise probabilities:

$$P(\mathbf{x}|c_i) = P(x_1, x_2, \dots, x_d|c_i) = \prod_{j=1}^d P(x_j|c_i)$$

The likelihood for class  $c_i$ , for dimension  $X_j$ , is given as

$$P(x_j|c_i) \propto f(x_j|\hat{\mu}_{ij}, \hat{\sigma}_{ij}^2) = \frac{1}{\sqrt{2\pi}\hat{\sigma}_{ij}} \exp\left\{-\frac{(x_j - \hat{\mu}_{ij})^2}{2\hat{\sigma}_{ij}^2}\right\}$$

where  $\hat{\mu}_{ij}$  and  $\hat{\sigma}_{ij}^2$  denote the estimated mean and variance for attribute  $X_j$ , for class  $c_i$ .



# Naive Bayes Classifier

## Numeric Attributes

The naive assumption corresponds to setting all the covariances to zero in  $\hat{\Sigma}_i$ , that is,

$$\Sigma_i = \begin{pmatrix} \sigma_{i1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{i2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & \sigma_{id}^2 \end{pmatrix}$$

The naive Bayes classifier thus uses the sample mean  $\hat{\mu}_i = (\hat{\mu}_{i1}, \dots, \hat{\mu}_{id})^T$  and a *diagonal* sample covariance matrix  $\hat{\Sigma}_i = \text{diag}(\sigma_{i1}^2, \dots, \sigma_{id}^2)$  for each class  $c_i$ . In total  $2d$  parameters have to be estimated, corresponding to the sample mean and sample variance for each dimension  $X_j$ .

# Naive Bayes Algorithm

**NaiveBayes** ( $D = \{(x_j, y_j)\}_{j=1}^n$ ):

```
1 for  $i = 1, \dots, k$  do
2    $D_i \leftarrow \{x_j^T \mid y_j = c_i, j = 1, \dots, n\}$  // class-specific subsets
3    $n_i \leftarrow |D_i|$  // cardinality
4    $\hat{P}(c_i) \leftarrow n_i/n$  // prior probability
5    $\hat{\mu}_i \leftarrow \frac{1}{n_i} \sum_{x_j \in D_i} x_j$  // mean
6    $\bar{D}_i = D_i - 1 \cdot \hat{\mu}_i^T$  // centered data for class  $c_i$ 
7   for  $j = 1, \dots, d$  do // class-specific var for  $j$ th attribute
8      $\hat{\sigma}_{ij}^2 \leftarrow \frac{1}{n_i} (\bar{X}_j^i)^T (\bar{X}_j^i)$  // variance
9    $\hat{\sigma}_i \leftarrow (\hat{\sigma}_{i1}^2, \dots, \hat{\sigma}_{id}^2)^T$  // class-specific attribute variances
10 return  $\hat{P}(c_i), \hat{\mu}_i, \hat{\sigma}_i$  for all  $i = 1, \dots, k$ 
```

**Testing** ( $x$  and  $\hat{P}(c_i), \hat{\mu}_i, \hat{\sigma}_i$ , for all  $i \in [1, k]$ ):

```
11  $\hat{y} \leftarrow \arg \max_{c_i} \left\{ \hat{P}(c_i) \prod_{j=1}^d f(x_j \mid \hat{\mu}_{ij}, \hat{\sigma}_{ij}^2) \right\}$ 
12 return  $\hat{y}$ 
```

## Iris Data: Naive Bayes

Consider 2D Iris Data consisting of sepal length and sepal width.  $\hat{P}(c_i)$  and  $\hat{\mu}_i$  remain unchanged.  $\hat{\Sigma}_i$  are assumed to be diagonal:

$$\hat{\Sigma}_1 = \begin{pmatrix} 0.1218 & 0 \\ 0 & 0.1423 \end{pmatrix} \quad \hat{\Sigma}_2 = \begin{pmatrix} 0.435 & 0 \\ 0 & 0.1096 \end{pmatrix}$$

Figure shows the contour or level curve (corresponding to 1% of the peak density) of the multivariate normal distribution for both classes. The diagonal assumption leads to contours that are axis-parallel ellipses.

For the test point  $\mathbf{x} = (6.75, 4.25)^T$ , the posterior probabilities for  $c_1$  and  $c_2$  are as follows:

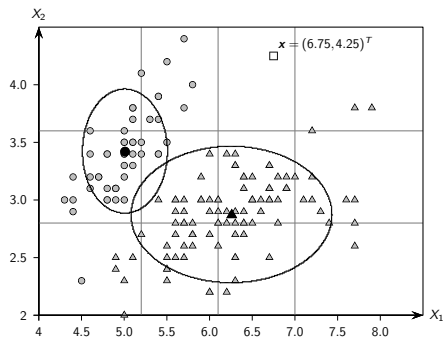
$$\hat{P}(c_1|\mathbf{x}) \propto \hat{f}(\mathbf{x}|\hat{\mu}_1, \hat{\Sigma}_1)\hat{P}(c_1) = (4.014 \times 10^{-7}) \times 0.33 = 1.325 \times 10^{-7}$$

$$\hat{P}(c_2|\mathbf{x}) \propto \hat{f}(\mathbf{x}|\hat{\mu}_2, \hat{\Sigma}_2)\hat{P}(c_2) = (9.585 \times 10^{-5}) \times 0.67 = 6.422 \times 10^{-5}$$

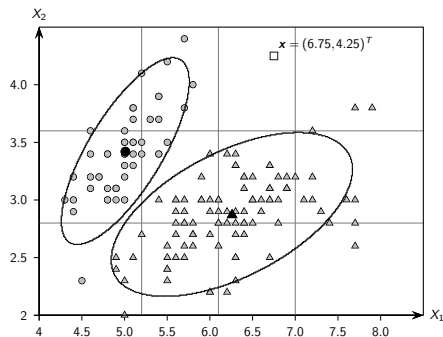
Because  $\hat{P}(c_2|\mathbf{x}) > \hat{P}(c_1|\mathbf{x})$ , it predicted  $\hat{y} = c_2$  to  $\mathbf{x}$ .

# Naive Bayes versus Full Bayes Classifier

$X_1$ :sepal length versus  $X_2$ :sepal width



(a) Naive Bayes



(b) Full Bayes

The independence assumption leads to a simplification:

$$P(\mathbf{x}|c_i) = \prod_{j=1}^d P(x_j|c_i) = \prod_{j=1}^d f(\mathbf{X}_j = \mathbf{e}_{jr_j} | c_i)$$

where  $f(\mathbf{X}_j = \mathbf{e}_{jr_j} | c_i)$  is the probability mass function for  $\mathbf{X}_j$ , estimated as:

$$\hat{f}(\mathbf{v}_j | c_i) = \frac{n_i(\mathbf{v}_j)}{n_i}$$

where  $n_i(\mathbf{v}_j)$  is the observed frequency of the value  $\mathbf{v}_j = \mathbf{e}_{jr_j}$  corresponding to the  $r_j$ th categorical value  $a_{jr_j}$  for the attribute  $X_j$  for class  $c_i$ .

If the count is zero, we can use the pseudo-count method to obtain a prior probability. The adjusted estimates with pseudo-counts are given as

$$\hat{f}(\mathbf{v}_j | c_i) = \frac{n_i(\mathbf{v}_j) + 1}{n_i + m_j}$$

where  $m_j = |\text{dom}(X_j)|$ .

# Discretized Iris Data: Naive Bayes

## Class-specific Empirical Joint Probability Mass Function

Class: $c_1$		$X_2$			$\hat{f}_{X_1}$
		Short ( $e_{21}$ )	Medium ( $e_{22}$ )	Long ( $e_{23}$ )	
$X_1$	Very Short ( $e_{11}$ )	1/50	33/50	5/50	39/50
	Short ( $e_{12}$ )	0	3/50	8/50	13/50
	Long ( $e_{13}$ )	0	0	0	0
	Very Long ( $e_{14}$ )	0	0	0	0
$\hat{f}_{X_2}$		1/50	36/50	13/50	

Class: $c_2$		$X_2$			$\hat{f}_{X_1}$
		Short ( $e_{21}$ )	Medium ( $e_{22}$ )	Long ( $e_{23}$ )	
$X_1$	Very Short ( $e_{11}$ )	6/100	0	0	6/100
	Short ( $e_{12}$ )	24/100	15/100	0	39/100
	Long ( $e_{13}$ )	13/100	30/100	0	43/100
	Very Long ( $e_{14}$ )	3/100	7/100	2/100	12/100
$\hat{f}_{X_2}$		46/100	52/100	2/100	

# Discretized Iris Data: Naive Bayes

The test point  $\mathbf{x} = (6.75, 4.25)$ , corresponding to (Long, Long) or  $\mathbf{v} = (\mathbf{e}_{13}, \mathbf{e}_{23})$ , is classified as follows:

$$\hat{P}(\mathbf{v} | c_1) = \hat{P}(\mathbf{e}_{13} | c_1) \cdot \hat{P}(\mathbf{e}_{23} | c_1) = \left( \frac{0 + 1}{50 + 4} \right) \cdot \left( \frac{13}{50} \right) = 4.81 \times 10^{-3}$$

$$\hat{P}(\mathbf{v} | c_2) = \hat{P}(\mathbf{e}_{13} | c_2) \cdot \hat{P}(\mathbf{e}_{23} | c_2) = \left( \frac{43}{100} \right) \cdot \left( \frac{2}{100} \right) = 8.60 \times 10^{-3}$$

$$\hat{P}(c_1 | \mathbf{v}) \propto (4.81 \times 10^{-3}) \times 0.33 = 1.59 \times 10^{-3}$$

$$\hat{P}(c_2 | \mathbf{v}) \propto (8.6 \times 10^{-3}) \times 0.67 = 5.76 \times 10^{-3}$$

Thus, the predicted class is  $\hat{y} = c_2$ .

# Nonparametric Approach

## $K$ Nearest Neighbors Classifier

We consider a non-parametric approach for likelihood estimation using the nearest neighbors density estimation.

Let  $\mathbf{D}$  be a training dataset comprising  $n$  points  $\mathbf{x}_i \in \mathbb{R}^d$ , and let  $\mathbf{D}_i$  denote the subset of points in  $\mathbf{D}$  that are labeled with class  $c_i$ , with  $n_i = |\mathbf{D}_i|$ .

Given a test point  $\mathbf{x} \in \mathbb{R}^d$ , and  $K$ , the number of neighbors to consider, let  $r$  denote the distance from  $\mathbf{x}$  to its  $K$ th nearest neighbor in  $\mathbf{D}$ .

Consider the  $d$ -dimensional hyperball of radius  $r$  around the test point  $\mathbf{x}$ , defined as

$$B_d(\mathbf{x}, r) = \{\mathbf{x}_i \in \mathbf{D} \mid \delta(\mathbf{x}, \mathbf{x}_i) \leq r\}$$

Here  $\delta(\mathbf{x}, \mathbf{x}_i)$  is the distance between  $\mathbf{x}$  and  $\mathbf{x}_i$ , which is usually assumed to be the Euclidean distance, i.e.,  $\delta(\mathbf{x}, \mathbf{x}_i) = \|\mathbf{x} - \mathbf{x}_i\|_2$ . We assume that  $|B_d(\mathbf{x}, r)| = K$ .



# Nonparametric Approach

## $K$ Nearest Neighbors Classifier

Let  $K_i$  denote the number of points among the  $K$  nearest neighbors of  $\mathbf{x}$  that are labeled with class  $c_i$ , that is

$$K_i = \{ \mathbf{x}_j \in B_d(\mathbf{x}, r) \mid y_j = c_i \}$$

The class conditional probability density at  $\mathbf{x}$  can be estimated as the fraction of points from class  $c_i$  that lie within the hyperball divided by its volume, that is

$$\hat{f}(\mathbf{x}|c_i) = \frac{K_i/n_i}{V} = \frac{K_i}{n_i V}$$

where  $V = \text{vol}(B_d(\mathbf{x}, r))$  is the volume of the  $d$ -dimensional hyperball.

The posterior probability  $P(c_i|\mathbf{x})$  can be estimated as

$$P(c_i|\mathbf{x}) = \frac{\hat{f}(\mathbf{x}|c_i)\hat{P}(c_i)}{\sum_{j=1}^k \hat{f}(\mathbf{x}|c_j)\hat{P}(c_j)}$$

However, because  $\hat{P}(c_i) = \frac{n_i}{n}$ , we have

$$\hat{f}(\mathbf{x}|c_i)\hat{P}(c_i) = \frac{K_i}{n_i V} \cdot \frac{n_i}{n} = \frac{K_i}{nV}$$

# Nonparametric Approach

## $K$ Nearest Neighbors Classifier

The posterior probability is given as

$$P(c_i|\mathbf{x}) = \frac{\frac{K_i}{nV}}{\sum_{j=1}^k \frac{K_j}{nV}} = \frac{K_i}{K}$$

Finally, the predicted class for  $\mathbf{x}$  is

$$\hat{y} = \arg \max_{c_i} \{P(c_i|\mathbf{x})\} = \arg \max_{c_i} \left\{ \frac{K_i}{K} \right\} = \arg \max_{c_i} \{K_i\}$$

Because  $K$  is fixed, the KNN classifier predicts the class of  $\mathbf{x}$  as the majority class among its  $K$  nearest neighbors.

# Iris Data

## $K$ Nearest Neighbors Classifier

Consider the 2D Iris dataset. Class  $c_1$  (circles) has  $n_1 = 50$  points and  $c_2$  (triangles) has  $n_2 = 100$  points.

We use KNN ( $K = 5$ ) to classify the test point  $\mathbf{x} = (6.75, 4.25)^T$ .

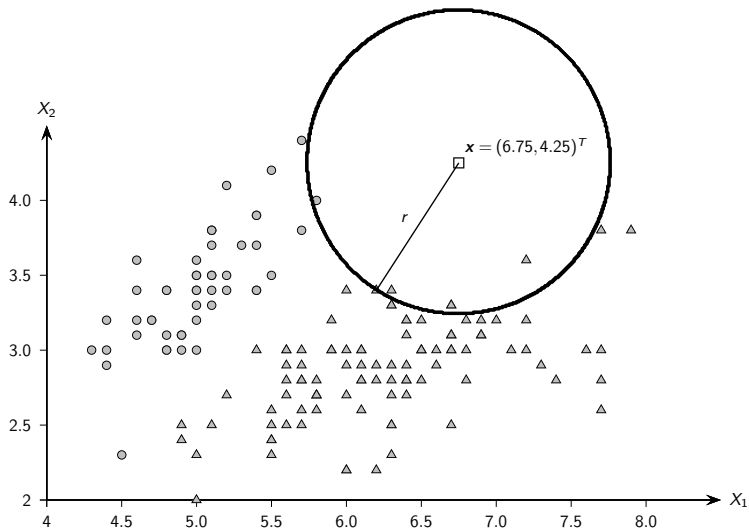
The distance from  $\mathbf{x}$  to its 5th nearest neighbor, namely  $(6.2, 3.4)^T$ , is given as  $r = \sqrt{1.025} = 1.012$ .

The enclosing ball or circle of radius  $r$  encompasses  $K_1 = 1$  point from  $c_1$  and  $K_2 = 4$  points from  $c_2$ .

Therefore, the predicted class for  $\mathbf{x}$  is  $\hat{y} = c_2$ .

# Iris Data

## $K$ Nearest Neighbors Classifier



# Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki<sup>1</sup>    Wagner Meira Jr.<sup>2</sup>

<sup>1</sup>Department of Computer Science  
Rensselaer Polytechnic Institute, Troy, NY, USA

<sup>2</sup>Department of Computer Science  
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 18: Probabilistic Classification