Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 18: Probabilistic Classification

Bayes Classifier

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

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

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

According to the Bayes theorem, we have

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

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

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

Estimating the Prior Probability: $P(c_i)$

Let $\boldsymbol{D}_i$ denote the subset of points in $\boldsymbol{D}$ that are labeled with class c_i:

$$\boldsymbol{D}_i = \{\boldsymbol{x}_j \in \boldsymbol{D} \mid \boldsymbol{x}_j \text{ has class } y_j = c_i\}$$

Let the size of the dataset $\boldsymbol{D}$ be given as $|\boldsymbol{D}| = n$, and let the size of each class-specific subset $\boldsymbol{D}_i$ be given as $|\boldsymbol{D}_i| = n_i$.

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

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 $\widehat{\boldsymbol{\Sigma}}_i$ to compute the probability density at $\boldsymbol{x}$

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

The posterior probability is then given as

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

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

$$\hat{y} = \arg\max_{c_i}\Big\{\hat{f}_i(\boldsymbol{x})P(c_i)\Big\}$$

Bayes Classifier Algorithm

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

for $i = 1, \ldots, k$ **do**

  $\boldsymbol{D}_i \leftarrow \{\boldsymbol{x}_j \mid y_j = c_i, j = 1, \ldots, n\}$ // class-specific subsets

  $n_i \leftarrow |\boldsymbol{D}_i|$ // cardinality

  $\hat{P}(c_i) \leftarrow n_i / n$ // prior probability

  $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\boldsymbol{x}_j \in \boldsymbol{D}_i} \boldsymbol{x}_j$ // mean

  $\boldsymbol{Z}_i \leftarrow \boldsymbol{D}_i - 1_{n_i} \hat{\boldsymbol{\mu}}_i^T$ // centered data

  $\widehat{\boldsymbol{\Sigma}}_i \leftarrow \frac{1}{n_i} \boldsymbol{Z}_i^T \boldsymbol{Z}_i$ // covariance matrix

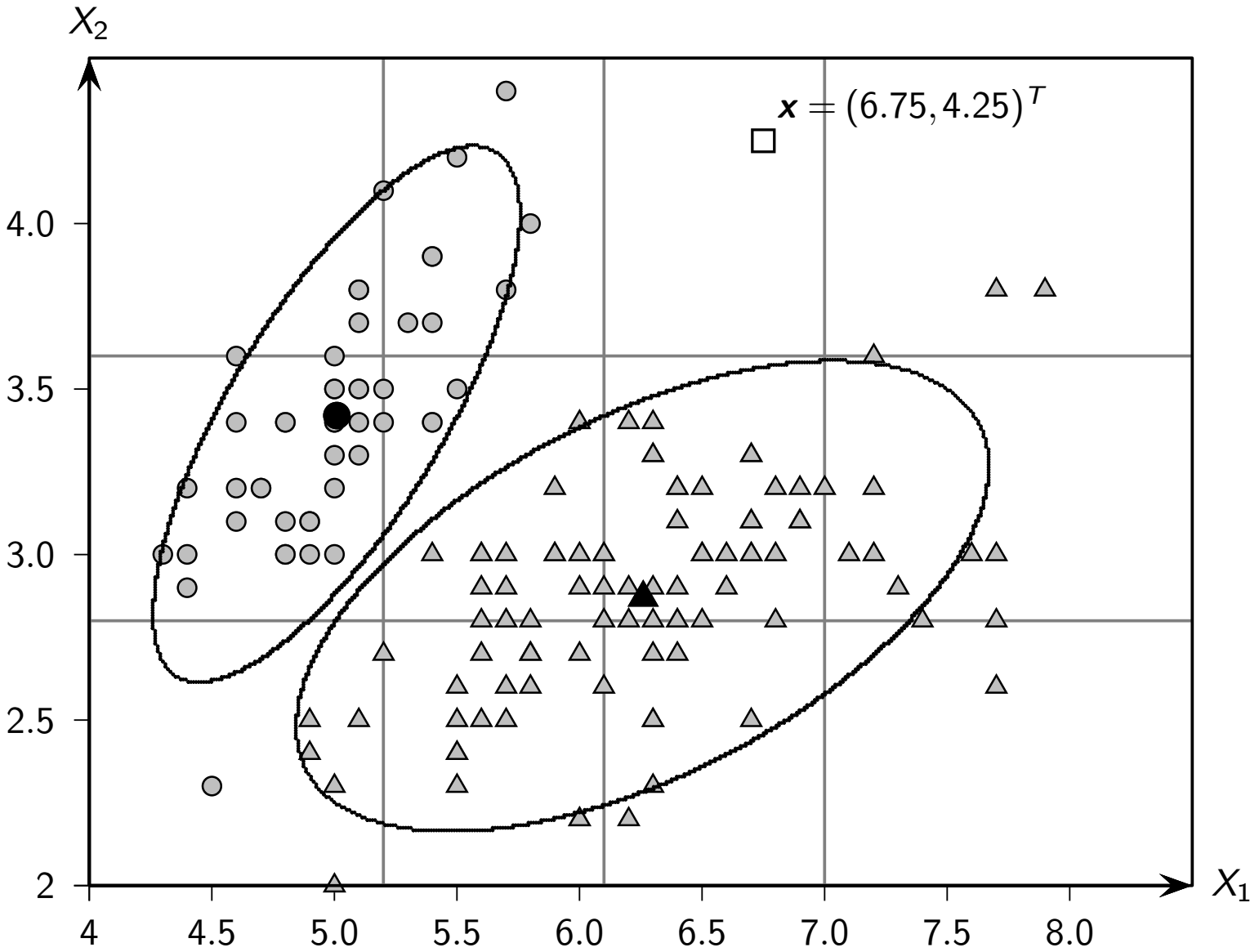
return $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i$ *for all* $i = 1, \ldots, k$

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

$\hat{y} \leftarrow \arg\max_{c_i} \left\{ f(\boldsymbol{x} | \hat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i) \cdot P(c_i) \right\}$ **return** $\hat{y}$

Bayes Classifier: Iris Data

X_1:sepal length versus X_2:sepal width



Bayes Classifier

Categorical Attributes

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

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

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

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

Bayes Classifier

Categorical Attributes

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

$$P(\boldsymbol{x}|c_i) = f(\boldsymbol{v}|c_i) = f\left(\boldsymbol{X}_1 = \boldsymbol{e}_{1r_1}, \ldots, \boldsymbol{X}_d = \boldsymbol{e}_{dr_d} | \; c_i\right)$$

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

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

where $n_i(\boldsymbol{v})$ is the number of times the value $\boldsymbol{v}$ occurs in class c_i.
However, to avoid zero probabilities we add a *pseudo-count* of 1 for each value

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

Discretized Iris Data

sepal length and sepal width

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

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	

Iris Data: Test Case

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

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

$$\hat{P}(\boldsymbol{x}|c_1) = \hat{f}(\boldsymbol{v}|c_1) = 3/50 = 0.06$$
$$\hat{P}(\boldsymbol{x}|c_2) = \hat{f}(\boldsymbol{v}|c_2) = 15/100 = 0.15$$
$$\hat{P}(c_1|\boldsymbol{x}) \propto 0.06 \times 0.33 = 0.0198$$
$$\hat{P}(c_2|\boldsymbol{x}) \propto 0.15 \times 0.67 = 0.1005$$

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

Iris Data: Test Case with Pseudo-counts

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

Unfortunately the probability mass at $\boldsymbol{v}$ 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

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

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

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

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

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 $\boldsymbol{v}$, 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 $\boldsymbol{v}$, 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:

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

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.

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

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

The naive Bayes classifier thus uses the sample mean $\hat{\boldsymbol{\mu}}_i = (\hat{\mu}_{i1}, \ldots, \hat{\mu}_{id})^T$ and a *diagonal* sample covariance matrix $\widehat{\Sigma}_i = diag(\sigma_{i1}^2, \ldots, \sigma_{id}^2)$ 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 ($\boldsymbol{D} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{n}$):

for $i = 1, \ldots, k$ **do**

  $\boldsymbol{D}_i \leftarrow \{\boldsymbol{x}_j \mid y_j = c_i, j = 1, \ldots, n\}$ // class-specific subsets

  $n_i \leftarrow |\boldsymbol{D}_i|$ // cardinality

  $\hat{P}(c_i) \leftarrow n_i / n$ // prior probability

  $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\boldsymbol{x}_j \in \boldsymbol{D}_i} \boldsymbol{x}_j$ // mean

  $\boldsymbol{Z}_i = \boldsymbol{D}_i - 1 \cdot \hat{\boldsymbol{\mu}}_i^T$ // centered data for class c_i

  for $j = 1, .., d$ **do** // class-specific variance for X_j

    $\hat{\sigma}_{ij}^2 \leftarrow \frac{1}{n_i} \boldsymbol{Z}_{ij}^T \boldsymbol{Z}_{ij}$ // variance

  $\hat{\boldsymbol{\sigma}}_i = \left(\hat{\sigma}_{i1}^2, \ldots, \hat{\sigma}_{id}^2\right)^T$ // class-specific attribute variances

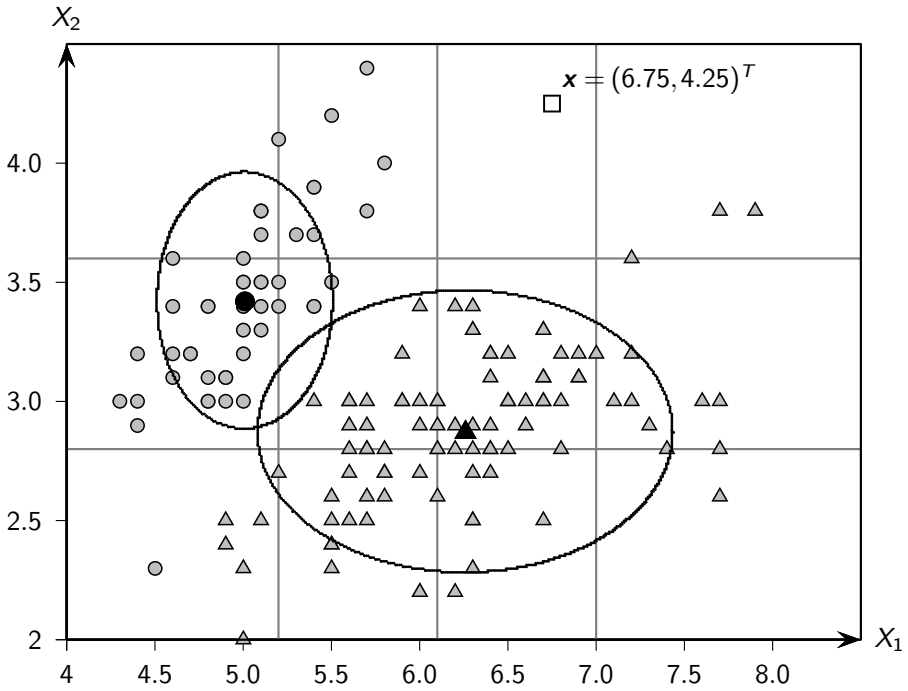
return $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i$ for all $i = 1, \ldots, k$

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

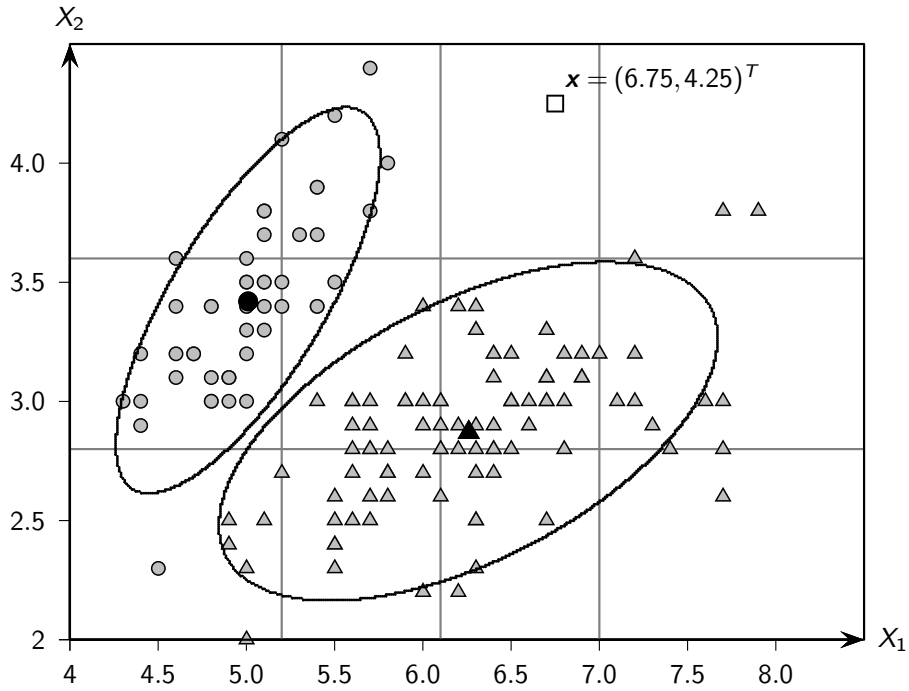
$\left(\qquad \right)^{d}$

Naive Bayes versus Full Bayes Classifier

X_1:sepal length versus X_2:sepal width



(a) Naive Bayes



(b) Full Bayes

Naive Bayes

Categorical Attributes

The independence assumption leads to a simplification of the joint probability mass function

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

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

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

where $n_i(\boldsymbol{v}_j)$ is the observed frequency of the value $\boldsymbol{v}_j = \boldsymbol{e}_j r_j$ corresponding to the r_jth 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

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

Nonparametric Approach

K Nearest Neighbors Classifier

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

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

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

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

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

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

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

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

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

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

where $V = \text{vol}(B_d(\boldsymbol{x}, r))$ is the volume of the d-dimensional hyperball.
The posterior probability $P(c_i|\boldsymbol{x})$ can be estimated as

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

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

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

The posterior probability is given as

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

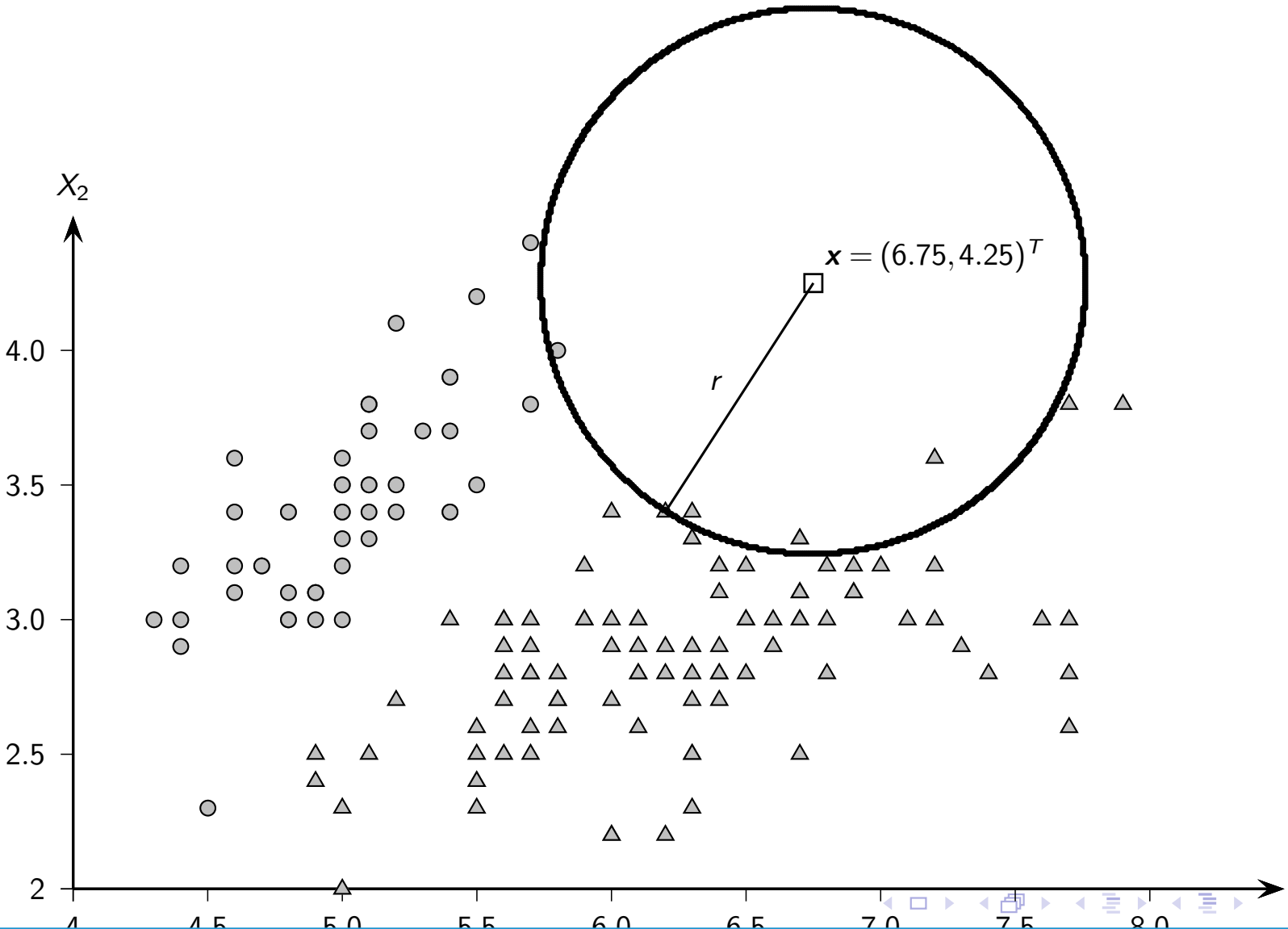
Finally, the predicted class for $\boldsymbol{x}$ is

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

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

Iris Data

K Nearest Neighbors Classifier



Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 18: Probabilistic Classification