CHAPTER 18 Probabilistic Classification

Classification refers to the task of predicting a class label for a given unlabeled point. In this chapter we consider three examples of the probabilistic classification approach. The (full) Bayes classifier uses the Bayes theorem to predict the class as the one that maximizes the posterior probability. The main task is to estimate the joint probability density function for each class, which is modeled via a multivariate normal distribution. The naive Bayes classifier assumes that attributes are independent, but it is still surprisingly powerful for many applications. We also describe the nearest neighbors classifier, which uses a non-parametric approach to estimate the density.

18.1 BAYES CLASSIFIER

Let the training dataset $\mathbf{D}$ consist of n points $\mathbf{x}_i$ in a d-dimensional space, and let y_i denote the class for each point, with $y_i \in \{c_1, c_2, \ldots, c_k\}$. The Bayes classifier directly uses the Bayes theorem to predict the class for a new test instance, $\mathbf{x}$. It 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

$$\hat{y} = \arg\max_{c_i}\{P(c_i|\mathbf{x})\} \tag{18.1}$$

The Bayes theorem allows us to invert the posterior probability in terms of the likelihood and prior probability, as follows:

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

where $P(\mathbf{x}|c_i)$ is the *likelihood*, defined as the probability of observing $\mathbf{x}$ assuming that the true class is c_i, $P(c_i)$ is the *prior probability* of class c_i, and $P(\mathbf{x})$ is the probability of observing $\mathbf{x}$ from any of the k classes, given as

$$P(\mathbf{x}) = \sum_{j=1}^{k} P(\mathbf{x}|c_j) \cdot P(c_j)$$

Because $P(\mathbf{x})$ is fixed for a given point, Bayes rule [Eq. (18.1)] can be rewritten as

$$\begin{aligned}\hat{y} &= \arg\max_{c_i}\{P(c_i|\mathbf{x})\} \\ &= \arg\max_{c_i}\left\{\frac{P(\mathbf{x}|c_i)P(c_i)}{P(\mathbf{x})}\right\} = \arg\max_{c_i}\big\{P(\mathbf{x}|c_i)P(c_i)\big\}\end{aligned} \tag{18.2}$$

In other words, the predicted class essentially depends on the likelihood of that class taking its prior probability into account.

18.1.1 Estimating the Prior Probability

To classify points, we have to estimate the likelihood and prior probabilities directly from the training dataset $\mathbf{D}$. Let $\mathbf{D}_i$ denote the subset of points in $\mathbf{D}$ that are labeled with class c_i:

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

Let the size of the dataset $\mathbf{D}$ be given as $|\mathbf{D}| = n$, and let the size of each class-specific subset $\mathbf{D}_i$ be given as $|\mathbf{D}_i| = n_i$. The prior probability for class c_i can be estimated as follows:

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

18.1.2 Estimating the Likelihood

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

Numeric Attributes

Assuming all dimensions are numeric, we can estimate the joint probability of $\mathbf{x}$ via either a nonparametric or a parametric approach. We consider the non-parametric approach in Section 18.3.

In the parametric approach we typically assume that each class c_i is normally distributed around some mean $\boldsymbol{\mu}_i$ with a corresponding covariance matrix $\boldsymbol{\Sigma}_i$, both of which are estimated from $\mathbf{D}_i$. For class c_i, the probability density at $\mathbf{x}$ is thus given as

$$f_i(\mathbf{x}) = f(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(\sqrt{2\pi})^d\sqrt{|\boldsymbol{\Sigma}_i|}}\exp\left\{-\frac{(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}{2}\right\} \tag{18.3}$$

Because c_i is characterized by a continuous distribution, the probability of any given point must be zero, i.e., $P(\mathbf{x}|c_i) = 0$. However, we can compute the likelihood by considering a small interval $\epsilon > 0$ centered at $\mathbf{x}$:

$$P(\mathbf{x}|c_i) = 2\epsilon \cdot f_i(\mathbf{x})$$

The posterior probability is then given as

$$P(c_i|\mathbf{x}) = \frac{2\epsilon \cdot f_i(\mathbf{x})P(c_i)}{\sum_{j=1}^{k} 2\epsilon \cdot f_j(\mathbf{x})P(c_j)} = \frac{f_i(\mathbf{x})P(c_i)}{\sum_{j=1}^{k} f_j(\mathbf{x})P(c_j)} \tag{18.4}$$

Further, because $\sum_{j=1}^{k} f_j(\mathbf{x})P(c_j)$ remains fixed for $\mathbf{x}$, we can predict the class for $\mathbf{x}$ by modifying Eq. (18.2) as follows:

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

To classify a numeric test point $\mathbf{x}$, the Bayes classifier estimates the parameters via the sample mean and sample covariance matrix. The sample mean for the class c_i can be estimated as

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathbf{D}_i} \mathbf{x}_j$$

and the sample covariance matrix for each class can be estimated using Eq. (2.30), as follows

$$\widehat{\boldsymbol{\Sigma}}_i = \frac{1}{n_i} \mathbf{Z}_i^T \mathbf{Z}_i$$

where $\mathbf{Z}_i$ is the centered data matrix for class c_i given as $\mathbf{Z}_i = \mathbf{D}_i - \mathbf{1} \cdot \hat{\boldsymbol{\mu}}_i^T$. These values can be used to estimate the probability density in Eq. (18.3) as $\hat{f}_i(\mathbf{x}) = f(\mathbf{x}|\hat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i)$.

Algorithm 18.1 shows the pseudo-code for the Bayes classifier. Given an input dataset $\mathbf{D}$, the method estimates the prior probability, mean and covariance matrix for each class. For testing, given a test point $\mathbf{x}$, it simply returns the class with the maximum posterior probability. The cost of training is dominated by the covariance matrix computation step which takes $O(nd^2)$ time.

ALGORITHM 18.1. Bayes Classifier

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

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

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

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

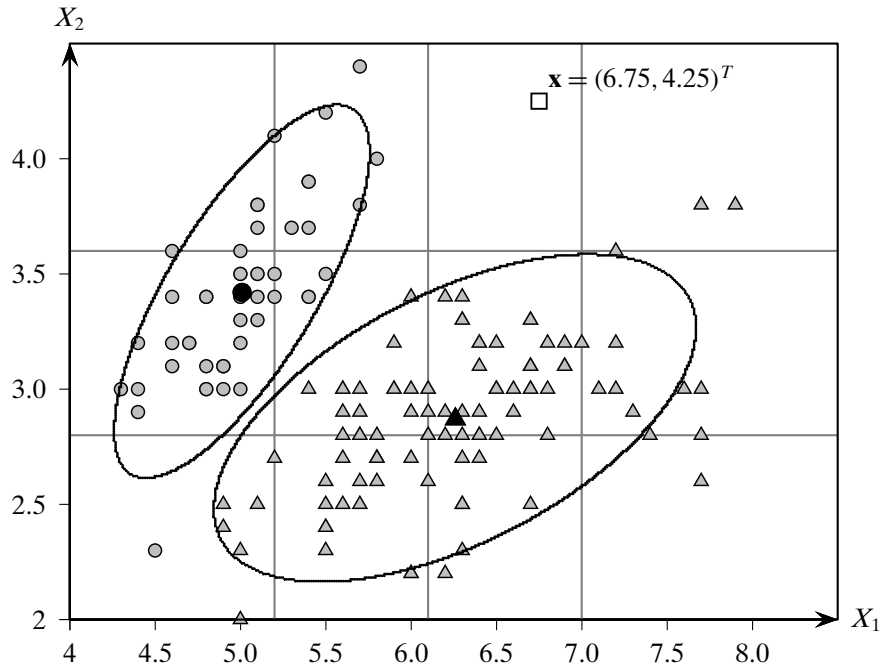

Figure 18.1. Iris data: X_1:`sepal length` versus X_2:`sepal width`. The class means are show in black; the density contours are also shown. The square represents a test point labeled $\mathbf{x}$.

Example 18.1. Consider the 2-dimensional Iris data, with attributes `sepal length` and `sepal width`, shown in Figure 18.1. Class c_1, which corresponds to `iris-setosa` (shown as circles), has $n_1 = 50$ points, whereas the other class c_2 (shown as triangles) has $n_2 = 100$ points. The prior probabilities for the two classes are

$$\hat{P}(c_1) = \frac{n_1}{n} = \frac{50}{150} = 0.33 \qquad \hat{P}(c_2) = \frac{n_2}{n} = \frac{100}{150} = 0.67$$

The means for c_1 and c_2 (shown as black circle and triangle) are given as

$$\hat{\boldsymbol{\mu}}_1 = \begin{pmatrix} 5.01 \\ 3.42 \end{pmatrix} \qquad \hat{\boldsymbol{\mu}}_2 = \begin{pmatrix} 6.26 \\ 2.87 \end{pmatrix}$$

and the corresponding covariance matrices are as follows:

$$\widehat{\boldsymbol{\Sigma}}_1 = \begin{pmatrix} 0.122 & 0.098 \\ 0.098 & 0.142 \end{pmatrix} \qquad \widehat{\boldsymbol{\Sigma}}_2 = \begin{pmatrix} 0.435 & 0.121 \\ 0.121 & 0.110 \end{pmatrix}$$

Figure 18.1 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.

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 using Eq. (18.4):

$$\hat{P}(c_1|\mathbf{x}) \propto \hat{f}(\mathbf{x}|\hat{\boldsymbol{\mu}}_1, \widehat{\boldsymbol{\Sigma}}_1)\hat{P}(c_1) = (4.914 \times 10^{-7}) \times 0.33 = 1.622 \times 10^{-7}$$

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

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

Categorical Attributes

If the attributes are categorical, the likelihood can be computed using the categorical data modeling approach presented in Chapter 3. Formally, let X_j be a categorical attribute over the domain $dom(X_j) = \{a_{j1}, a_{j2}, \ldots, a_{jm_j}\}$, that is, attribute X_j can take on m_j distinct categorical values. 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 $\mathbf{e}_{j1}, \mathbf{e}_{j2}, \ldots, \mathbf{e}_{jm_j}$, where $\mathbf{e}_{jr}$ 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 $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_d)^T$. Let $d' = \sum_{j=1}^{d} m_j$; a categorical point $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$ 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_jth value in the domain of X_j. The probability of the categorical point $\mathbf{x}$ is obtained from the joint probability mass function (PMF) for the vector random variable $\mathbf{X}$:

$$P(\mathbf{x}|c_i) = f(\mathbf{v}|c_i) = f\big(\mathbf{X}_1 = \mathbf{e}_{1r_1}, \ldots, \mathbf{X}_d = \mathbf{e}_{dr_d} | \, c_i\big) \tag{18.5}$$

The above 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. Unfortunately, if the probability mass at the point $\mathbf{v}$ is zero for one or both classes, it would lead to a zero value for the posterior probability. To avoid zero probabilities, one approach is to introduce a small prior probability for all the possible values of the vector random variable $\mathbf{X}$. One simple approach is to assume a *pseudo-count* of 1 for each value, that is, to assume that each value of $\mathbf{X}$ occurs at least one time, and to augment this base count of 1 with the actual number of occurrences of the observed value $\mathbf{v}$ in class c_i. The adjusted probability mass at $\mathbf{v}$ is then given as

$$\hat{f}(\mathbf{v}|c_i) = \frac{n_i(\mathbf{v}) + 1}{n_i + \prod_{j=1}^{d} m_j} \tag{18.6}$$

where $\prod_{j=1}^{d} m_j$ gives the number of possible values of $\mathbf{X}$. Extending the code in Algorithm 18.1 to incorporate categorical attributes is relatively straightforward; all that is required is to compute the joint PMF for each class using Eq. (18.6).

Table 18.1. Discretized `sepal length` and `sepal width` attributes

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`

Table 18.2. Class-specific empirical (joint) probability mass function

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

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

Example 18.2. Assume that the `sepal length` and `sepal width` attributes in the Iris dataset have been discretized as shown in Table 18.1a and Table 18.1b, respectively. We have $|dom(X_1)| = m_1 = 4$ and $|dom(X_2)| = m_2 = 3$. These intervals are also illustrated in Figure 18.1: via the gray grid lines. Table 18.2 shows the empirical joint PMF for both the classes. Also, as in Example 18.1, the prior probabilities of the classes are given as $\hat{P}(c_1) = 0.33$ and $\hat{P}(c_2) = 0.67$.

Consider a test point $\mathbf{x} = (5.3, 3.0)^T$ corresponding to the categorical point (`Short, Medium`), which is represented as $\mathbf{v} = \begin{pmatrix} \mathbf{e}_{12}^T & \mathbf{e}_{22}^T \end{pmatrix}^T$. The likelihood and posterior probability for each class is given as

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

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

On the other hand, the test point $\mathbf{x} = (6.75, 4.25)^T$ corresponding to the categorical point (`Long, Long`) is represented as $\mathbf{v} = \begin{pmatrix} \mathbf{e}_{13}^T & \mathbf{e}_{23}^T \end{pmatrix}^T$. Unfortunately the

probability mass at $\mathbf{v}$ is zero for both classes. We adjust the PMF via pseudo-counts [Eq. (18.6)]; note 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 $\hat{y} = c_2$.

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 instance, 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. To address some of these concerns we can use reduced set of parameters in practice, as described next.

18.2 NAIVE BAYES CLASSIFIER

We saw earlier that the full Bayes approach is fraught with estimation related problems, especially with large number of dimensions. The naive Bayes approach makes the simple assumption that all the attributes are independent. This leads to a much simpler, though surprisingly effective classifier in practice. The independence assumption immediately implies that the likelihood can be decomposed into a product of dimension-wise probabilities:

$$P(\mathbf{x}|c_i) = P(x_1, x_2, \ldots, x_d|c_i) = \prod_{j=1}^{d} P(x_j|c_i) \tag{18.7}$$

Numeric Attributes

For numeric attributes we make the default assumption that each of them is normally distributed for each class c_i. Let μ_{ij} and σ_{ij}^2 denote the mean and variance for attribute X_j, for class c_i. The likelihood for class c_i, for dimension X_j, is given as

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

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

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

This yields

$$|\mathbf{\Sigma}_i| = \det(\mathbf{\Sigma}_i) = \sigma_{i1}^2 \sigma_{i2}^2 \cdots \sigma_{id}^2 = \prod_{j=1}^{d} \sigma_{ij}^2$$

Also, we have

$$\mathbf{\Sigma}_i^{-1} = \begin{pmatrix} \frac{1}{\sigma_{i1}^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_{i2}^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & \frac{1}{\sigma_{id}^2} \end{pmatrix}$$

assuming that $\sigma_{ij}^2 \neq 0$ for all j. Finally,

$$(\mathbf{x} - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) = \sum_{j=1}^{d} \frac{(x_j - \mu_{ij})^2}{\sigma_{ij}^2}$$

Plugging these into Eq. (18.3) gives us

$$\begin{aligned} P(\mathbf{x}|c_i) &= \frac{1}{(\sqrt{2\pi})^d \sqrt{\prod_{j=1}^{d} \sigma_{ij}^2}} \exp\left\{ -\sum_{j=1}^{d} \frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2} \right\} \\ &= \prod_{j=1}^{d} \left(\frac{1}{\sqrt{2\pi}\, \sigma_{ij}} \exp\left\{ -\frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2} \right\} \right) \\ &= \prod_{j=1}^{d} P(x_j|c_i) \end{aligned}$$

which is equivalent to Eq. (18.7). In other words, the joint probability has been decomposed into a product of the probability along each dimension, as required by the independence assumption.

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

Algorithm 18.2 shows the pseudo-code for the naive Bayes classifier. Given an input dataset $\mathbf{D}$, the method estimates the prior probability and mean for each class. Next, it computes the variance $\hat{\sigma}_{ij}^2$ for each of the attributes X_j, with all the d variances for class c_i stored in the vector $\hat{\boldsymbol{\sigma}}_i$. The variance for attribute X_j is obtained by first

ALGORITHM 18.2. Naive Bayes Classifier

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

for $i = 1, \ldots, k$ **do**
  $\mathbf{D}_i \leftarrow \{\mathbf{x}_j \mid y_j = c_i, j = 1, \ldots, n\}$ // class-specific subsets
  $n_i \leftarrow |\mathbf{D}_i|$ // cardinality
  $\hat{P}(c_i) \leftarrow n_i / n$ // prior probability
  $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathbf{D}_i} \mathbf{x}_j$ // mean
  $\mathbf{Z}_i = \mathbf{D}_i - \mathbf{1} \cdot \hat{\boldsymbol{\mu}}_i^T$ // centered data for class c_i
  for $j = 1, .., d$ **do** // class-specific variance for X_j
    $\hat{\sigma}_{ij}^2 \leftarrow \frac{1}{n_i} Z_{ij}^T Z_{ij}$ // 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 (x and $\hat{P}(c_i)$, $\hat{\boldsymbol{\mu}}_i$, $\hat{\boldsymbol{\sigma}}_i$, for all $i \in [1, k]$):

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

return $\hat{y}$

centering the data for class $\mathbf{D}_i$ via $\mathbf{Z}_i = \mathbf{D}_i - \mathbf{1} \cdot \hat{\boldsymbol{\mu}}_i^T$. We denote by Z_{ij} the centered data for class c_i corresponding to attribute X_j. The variance is then given as $\hat{\sigma} = \frac{1}{n_i} Z_{ij}^T Z_{ij}$.

Training the naive Bayes classifier is very fast, with $O(nd)$ computational complexity. For testing, given a test point $\mathbf{x}$, it simply returns the class with the maximum posterior probability obtained as a product of the likelihood for each dimension and the class prior probability.

Example 18.3. Consider Example 18.1. In the naive Bayes approach the prior probabilities $\hat{P}(c_i)$ and means $\hat{\boldsymbol{\mu}}_i$ remain unchanged. The key difference is that the covariance matrices are assumed to be diagonal, as follows:

$$\widehat{\boldsymbol{\Sigma}}_1 = \begin{pmatrix} 0.122 & 0 \\ 0 & 0.142 \end{pmatrix} \qquad \widehat{\boldsymbol{\Sigma}}_2 = \begin{pmatrix} 0.435 & 0 \\ 0 & 0.110 \end{pmatrix}$$

Figure 18.2 shows the contour or level curve (corresponding to 1% of the peak density) of the multivariate normal distribution for both classes. One can see that the diagonal assumption leads to contours that are axis-parallel ellipses; contrast these with the contours in Figure 18.1 for the full Bayes classifier.

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

$$\hat{P}(c_1|\mathbf{x}) \propto \hat{f}(\mathbf{x}|\hat{\boldsymbol{\mu}}_1, \widehat{\boldsymbol{\Sigma}}_1)\hat{P}(c_1) = (3.99 \times 10^{-7}) \times 0.33 = 1.32 \times 10^{-7}$$

$$\hat{P}(c_2|\mathbf{x}) \propto \hat{f}(\mathbf{x}|\hat{\boldsymbol{\mu}}_2, \widehat{\boldsymbol{\Sigma}}_2)\hat{P}(c_2) = (9.597 \times 10^{-5}) \times 0.67 = 6.43 \times 10^{-5}$$

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

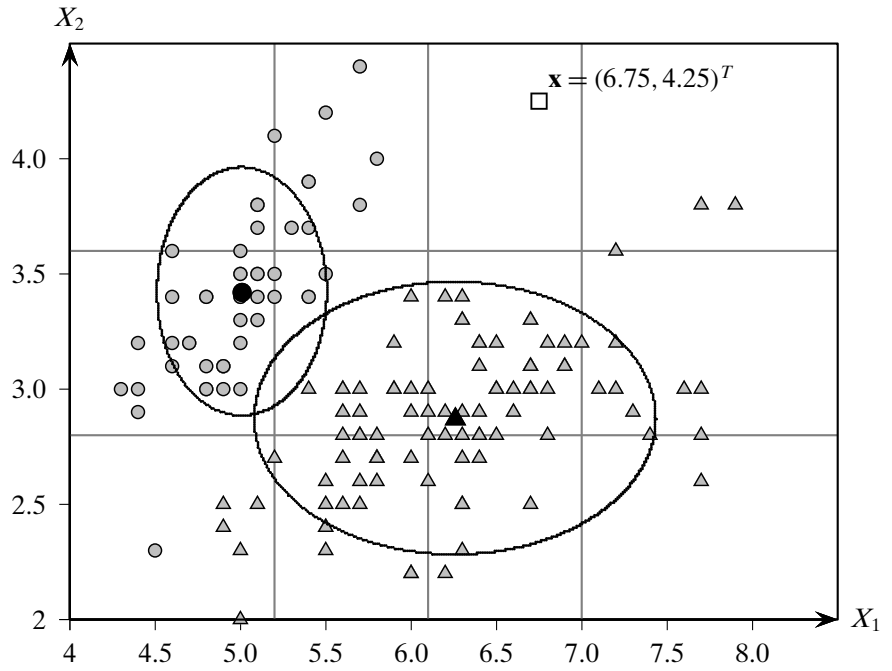

Figure 18.2. Naive Bayes: X_1:`sepal length` versus X_2:`sepal width`. The class means are shown in black; the density contours are also shown. The square represents a test point labeled $\mathbf{x}$.

Categorical Attributes

The independence assumption leads to a simplification of the joint probability mass function in Eq. (18.5), which can be rewritten as

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

where $f(\mathbf{X}_j = \mathbf{e}_{jr_j}|c_i)$ is the probability mass function for $\mathbf{X}_j$, which can be estimated from $\mathbf{D}_i$ as follows:

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

where $n_i(\mathbf{v}_j)$ is the observed frequency of the value $\mathbf{v}_j = \mathbf{e}_j r_j$ corresponding to the r_jth categorical value a_{jr_j} for the attribute X_j for class c_i. As in the full Bayes case, 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 $m_j = |dom(X_j)|$. Extending the code in Algorithm 18.2 to incorporate categorical attributes is straightforward.

Example 18.4. Continuing Example 18.2, the class-specific PMF for each discretized attribute is shown in Table 18.2. In particular, these correspond to the row and column marginal probabilities $\hat{f}_{X_1}$ and $\hat{f}_{X_2}$, respectively.

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 $\hat{y} = c_2$.

18.3 K NEAREST NEIGHBORS CLASSIFIER

In the preceding sections we considered a parametric approach for estimating the likelihood $P(\mathbf{x}|c_i)$. In this section, we consider a non-parametric approach, which does not make any assumptions about the underlying joint probability density function. Instead, it directly uses the data sample to estimate the density, for example, using the density estimation methods from Chapter 15. We illustrate the non-parametric approach using nearest neighbors density estimation from Section 15.2.3, which leads to the *K nearest neighbors* (KNN) classifier.

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 Kth 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., $\delta(\mathbf{x}, \mathbf{x}_i) = \|\mathbf{x} - \mathbf{x}_i\|_2$. However, other distance metrics can also be used. We assume that $|B_d(\mathbf{x}, r)| = K$.

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 [Eq. (6.4)].

Using Eq. (18.4), 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

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

Thus 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.

Example 18.5. Consider the 2D Iris dataset shown in Figure 18.3. The two classes are: c_1 (circles) with $n_1 = 50$ points and c_2 (triangles) with $n_2 = 100$ points.

Let us classify the test point $\mathbf{x} = (6.75, 4.25)^T$ using its $K = 5$ nearest neighbors. 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 is shown in the figure. It encompasses $K_1 = 1$ point from class c_1 and $K_2 = 4$ points from class c_2. Therefore, the predicted class for $\mathbf{x}$ is $\hat{y} = c_2$.

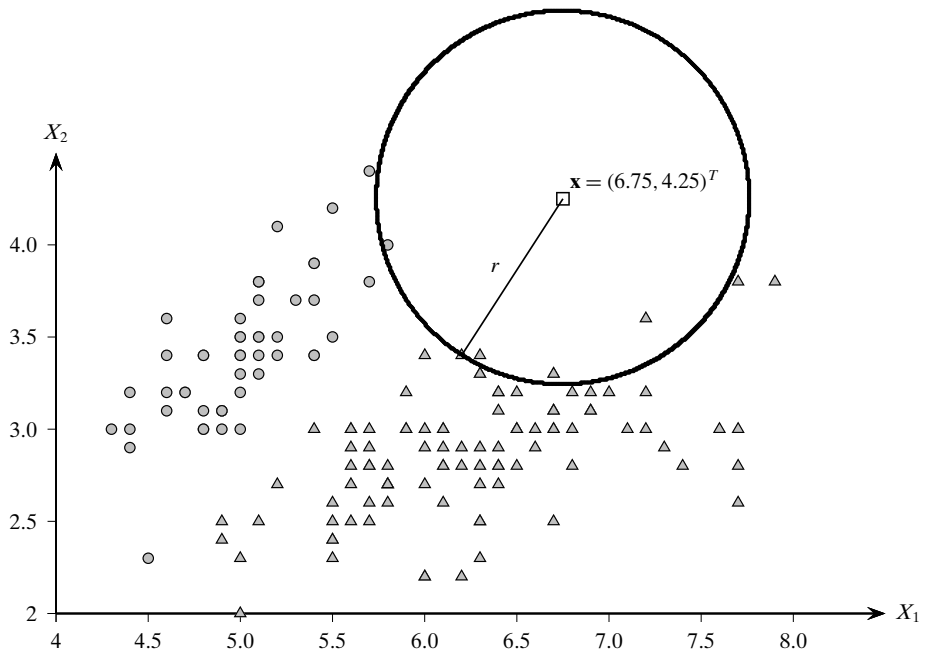

Figure 18.3. Iris Data: K Nearest Neighbors Classifier

18.4 FURTHER READING

The naive Bayes classifier is surprisingly effective even though the independence assumption is usually violated in real datasets. Comparison of the naive Bayes classifier against other classification approaches and reasons for why is works well have appeared in Langley, Iba, and Thompson (1992); Domingos and Pazzani (1997); Zhang (2005); Hand and Yu (2001) and Rish (2001). For the long history of naive Bayes in information retrieval see Lewis (1998). The K nearest neighbor classification approach was first proposed in Fix and Hodges, Jr. (1951).

Domingos, P. and Pazzani, M. (1997). "On the optimality of the simple Bayesian classifier under zero-one loss." *Machine Learning*, 29 (2–3): 103–130.

Fix, E. and Hodges Jr., J. L. (1951). Discriminatory analysis, nonparametric discrimination. *USAF School of Aviation Medicine, Randolph Field, TX, Project 21-49-004, Report 4, Contract AF41(128)-31*.

Hand, D. J. and Yu, K. (2001). "Idiot's Bayes-not so stupid after all?" *International Statistical Review*, 69 (3): 385–398.

Langley, P., Iba, W., and Thompson, K. (1992). "An analysis of Bayesian classifiers." *In Proceedings of the National Conference on Artificial Intelligence*, pp. 223–223.

Lewis, D. D. (1998). "Naive (Bayes) at forty: The independence assumption in information retrieval." *In Proceedings of the 10th European Conference on Machine Learning.* pp. 4–15.

Rish, I. (2001). "An empirical study of the naive Bayes classifier." *In Proceedings of the IJCAI Workshop on Empirical Methods in Artificial Intelligence*, pp. 41–46.

Zhang, H. (2005). "Exploring conditions for the optimality of naive Bayes." *International Journal of Pattern Recognition and Artificial Intelligence*, 19 (2): 183–198.

18.5 EXERCISES

Q1. Consider the dataset in Table 18.3. Classify the new point: (Age=23, Car=truck) via the full and naive Bayes approach. You may assume that the domain of Car is given as {sports, vintage, suv, truck}.

Table 18.3. Data for Q1

$\mathbf{x}_i$	Age	Car	Class
$\mathbf{x}_1$	25	sports	L
$\mathbf{x}_2$	20	vintage	H
$\mathbf{x}_3$	25	sports	L
$\mathbf{x}_4$	45	suv	H
$\mathbf{x}_5$	20	sports	H
$\mathbf{x}_6$	25	suv	H

Q2. Given the dataset in Table 18.4, use the naive Bayes classifier to classify the new point $(T, F, 1.0)$.

Table 18.4. Data for Q2

$\mathbf{x}_i$	a_1	a_2	a_3	Class
$\mathbf{x}_1$	T	T	5.0	Y
$\mathbf{x}_2$	T	T	7.0	Y
$\mathbf{x}_3$	T	F	8.0	N
$\mathbf{x}_4$	F	F	3.0	Y
$\mathbf{x}_5$	F	T	7.0	N
$\mathbf{x}_6$	F	T	4.0	N
$\mathbf{x}_7$	F	F	5.0	N
$\mathbf{x}_8$	T	F	6.0	Y
$\mathbf{x}_9$	F	T	1.0	N

Q3. Consider the class means and covariance matrices for classes c_1 and c_2:

$$\boldsymbol{\mu}_1 = (1,3) \qquad \boldsymbol{\mu}_2 = (5,5)$$

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 5 & 3 \\ 3 & 2 \end{pmatrix} \qquad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

Classify the point $(3,4)^T$ via the (full) Bayesian approach, assuming normally distributed classes, and $P(c_1) = P(c_2) = 0.5$. Show all steps. Recall that the inverse of a 2×2 matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is given as $A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$.