

3. Probabilistic models

3.1 Turning data into probabilities

Let us look at Fig. 3.1 which shows the measurements of variable X as probabilities (can also be interpreted as frequencies) for two classes c_1 and c_2 . The correct class is easy to predict at the beginning and end of the distribution, but the middle is unclear for the sake of some overlap.

Suppose that the purpose is to classify writing of letters 'a' and 'b' as in Fig. 3.2 based on their height. Usually, 'a' is written as a smaller letter than 'b' by most people, but not by everybody. Fortunately, we know that 'a' is much more common than 'b' in English text. If we see a letter either 'a' or 'b', there is a 75% chance that it is 'a'.

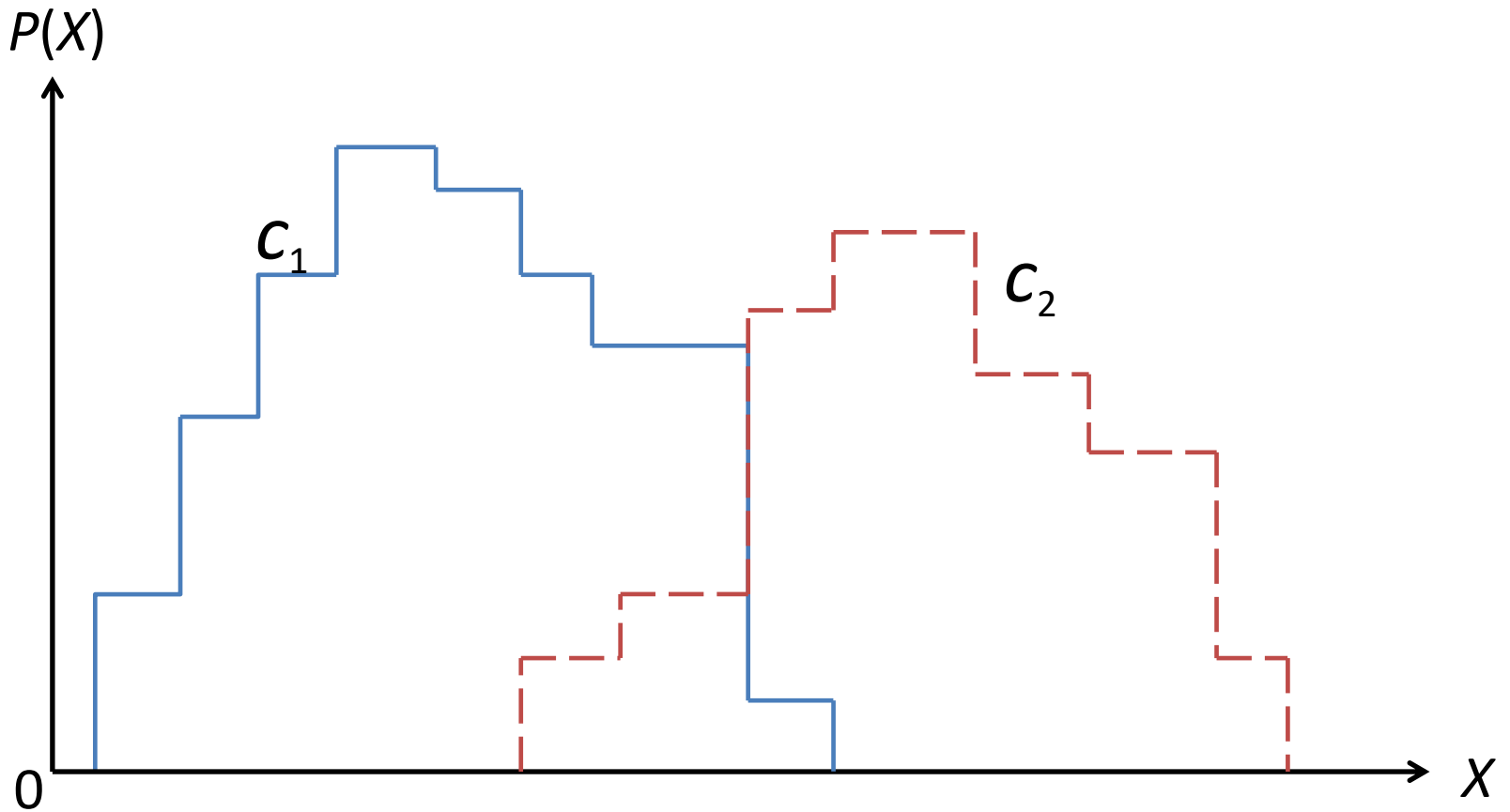


Fig 3.1 A histogram (frequency) of variable values x against their probability for two classes c_1 and c_2 .

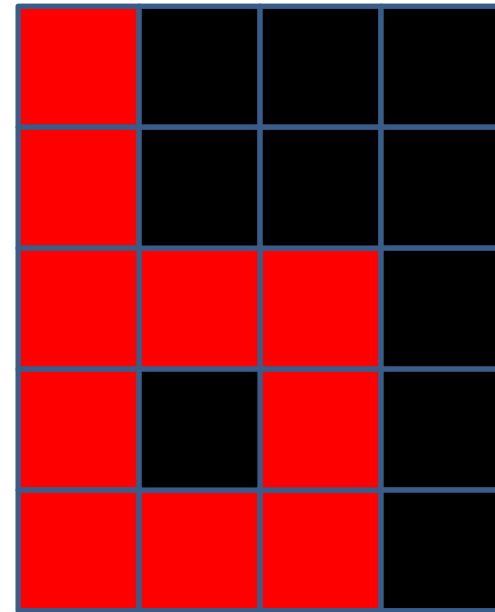
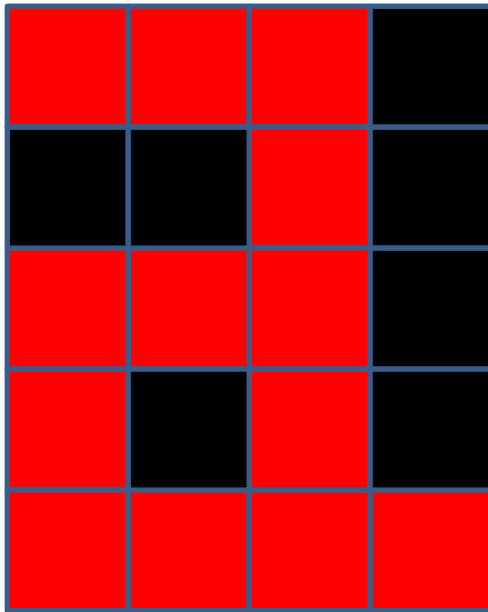


Fig. 3.2 Letters 'a' and 'b' in pixel form.

We are using prior (a priori) knowledge to estimate the probability that the letter is an 'a': in this example $P(c_1) = 0.75$, $P(c_2) = 0.25$.

Class probability $P(c_1)$ does not suffice, but we also use the values of X in a training set and the class that each case belongs to. This lets us calculate the value of $P(c_1)$ by counting how many times out of the total the class was c_1 and by dividing by the total number of cases. Then the conditional probability of c_1 given that variable X has value x is $P(c_1 | x)$. In Fig. 3.1 $P(c_1 | x)$ is much larger for small values of X than for large values. How do we get this conditional probability?

First we have to *quantize* or *discretize* the measurement x , which just means that we put it into one of discrete set of values $\{x\}$, such as the bins or intervals in a histogram. This is that is plotted in Fig. 3.1. If there are lots of cases in the two classes and the histogram intervals that their measurements fall into, we can compute *joint probability* $P(c_i, x_j)$ which shows how frequently a measurement of c_i fell into histogram interval x_j . According to interval x_j , we count the number of cases of class c_i that are in it, and divide by the total number of cases (of any class).

We can also define $P(x_j|c_i)$, which is a different conditional probability, and tells us how frequently (in the training set) there is a measurement of x_j given that the case is from class c_i . Again, we get this information from the histogram by counting the number of cases of class c_i in histogram interval x_j and dividing by the number of cases of that class (in any interval).

Finally, *Bayes's rule* is used, which is derived. There is a connection between the joint probability and the conditional probability

$$P(c_i, x_j) = P(x_j|c_i)P(c_i)$$

or equivalently

$$P(c_i, x_j) = P(c_i|x_j)P(x_j).$$

No doubt, the right-hand side of these two equations must be equal to each other, since they are equal to $P(c_i, x_j)$. With one division we get

$$P(c_i|x_j) = \frac{P(x_j|c_i)P(c_i)}{P(x_j)} \quad (3.1)$$

which is Bayes's rule. It relates the *posterior* (*a posteriori*) probability $P(c_i|x_j)$ with the *prior* (*a priori*) probability $P(c_i)$ and *class-conditional* probability $P(x_j|c_i)$.

The formula can now be written in the form:

$$P(c_i | x_j) = \frac{P(x_j | c_i)P(c_i)}{\sum_{k=1}^C P(x_j | c_k)P(c_k)} \quad (3.2)$$

Sometimes the following terms are used:

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

The denominator of the preceding formula (3.2) acts to normalize everything, so that all the probabilities sum to 1. Note that the denominator is the same value all the time when we compute probabilities $P(c_i|x_j)$ for different classes $i=1,2,\dots,C$. (Since it is constant, it can be reduced in association with such inequalities as (3.3) later on p. 91.)

Because any observation or case x_j has to belong to some class c_i , then we could marginalise in (3.2) over the classes to compute:

$$P(x_j) = \sum_{k=1}^C P(x_j|c_k)P(c_k)$$

The prior probabilities can be estimated by counting how frequently each class appears in the training set. The class-conditional probabilities are obtained from the frequencies of the values of the variable for the training set.

The posterior probability (Fig. 3.3) is used to assign each new case to one of the classes by picking the class c_i where:

$$P(c_i|\mathbf{x}) > P(c_j|\mathbf{x}) \quad \forall i \neq j \quad (3.3)$$

Here \mathbf{x} is a vector of values of all variables instead of one variable only as in those preceding formulas. This is called the *maximum a posteriori* (MAP) hypothesis. It gives us a way to choose which class to select as the output one.

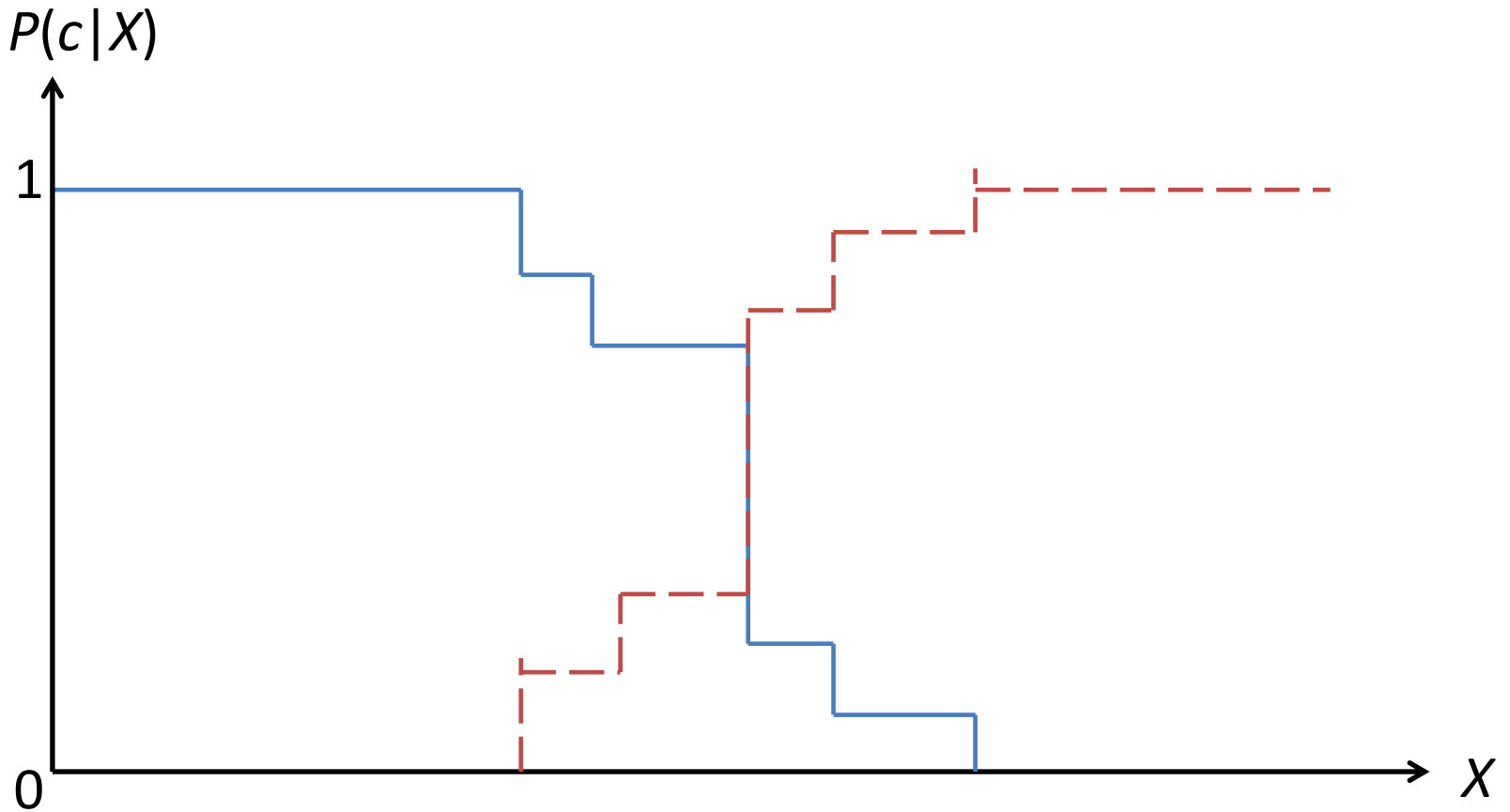


Fig. 3.3 The posterior probabilities of two classes c_1 and c_2 for variable X .

There is the MAP question: what is the most probable class given the training data? Let us suppose that there are three classes and for a particular input the posterior probabilities of the classes are $P(c_1|x)=0.35$, $P(c_2|x)=0.45$ and $P(c_3|x)=0.20$. The MAP hypothesis therefore expresses that this input is in class c_2 , because that is the class of the highest posterior probability. If the class is c_1 or c_3 , the action 1 is done, and if it is c_2 , action 2 is done.

Let us assume that the inputs are results of a blood test, three classes are different possible diseases, and the output is whether or not to treat with a particular antibiotic.

The MAP method has expressed that the output is c_2 and so the disease is not treated with a particular antibiotic.

What is the probability that it does not belong to class c_2 , and so should have been treated with the antibiotic? It is $1 - P(c_2) = 0.55$. Thus the MAP prediction seems to be wrong. We ought to treat, because overall it is more probable.

3.2 Minimising risk

In the medical example we saw how it made sense to classify based on minimizing the probability of misclassification. The *risk* from misclassifying someone unhealthy when they are healthy is usually smaller than the other way around, but not necessarily always. In a study⁵ classification was studied for two data sets collected for diagnosis of acute appendicitis in Finland and Germany. See Table 3.1. Here false negatives would be dangerous, since for actual cases a section (appendectomy) is typically made rapidly. False positives, if unnecessary sections made, are not so bad, but all sections are risky in principle and produce costs.

⁵E. Pesonen, Chr. Ohmann, M. Eskelinen and M. Juhola: Diagnosis of acute appendicitis in two databases. Evaluation of different neighbourhoods with an LVQ neural network, *Methods of Information in Medicine*, 37, 59-63, 1998.

Example 1

Table 3.1 Sensitivity (true positive rate) and specificity (true negative rate) of different training set and test set pairs with Learning Vector Quantization (LVQ) for the diagnosis of acute appendicitis: three classes (acute appendicitis one of them), 16 variables and 1846 patients (no missing values) altogether.

| Training set | Test set | Sensitivity | Specificity |
|--------------|----------|-------------|-------------|
| Finnish | Finnish | 0.84 | 0.89 |
| | German | 0.41 | 0.93 |
| | Mixed | 0.64 | 0.91 |
| German | Finnish | 0.91 | 0.71 |
| | German | 0.64 | 0.83 |
| | Mixed | 0.79 | 0.76 |
| Mixed | Finnish | 0.88 | 0.84 |
| | German | 0.47 | 0.93 |
| | Mixed | 0.69 | 0.88 |

Classification error

We can look at classification error based on

$$\begin{aligned} P(\text{classification error}) &= P(\mathbf{x} \in R_1, c_2) + P(\mathbf{x} \in R_2, c_1) \\ &= P(\mathbf{x} \in R_1 | c_2)P(c_2) + P(\mathbf{x} \in R_2 | c_1)P(c_1) \end{aligned}$$

where R_1 and R_2 are from Fig. 3.4. Let us hop for a while on continuous presentation (where we also use probability density p).

Using Bayes's theorem (rule), we find:

$$P(\text{classification error}) = \int_{R_1} p(\mathbf{x} | c_2)P(c_2)d\mathbf{x} + \int_{R_2} p(\mathbf{x} | c_1)P(c_1)d\mathbf{x}$$

We could generalize this for C classes, but it is more convenient to define the probability for the correct classification.

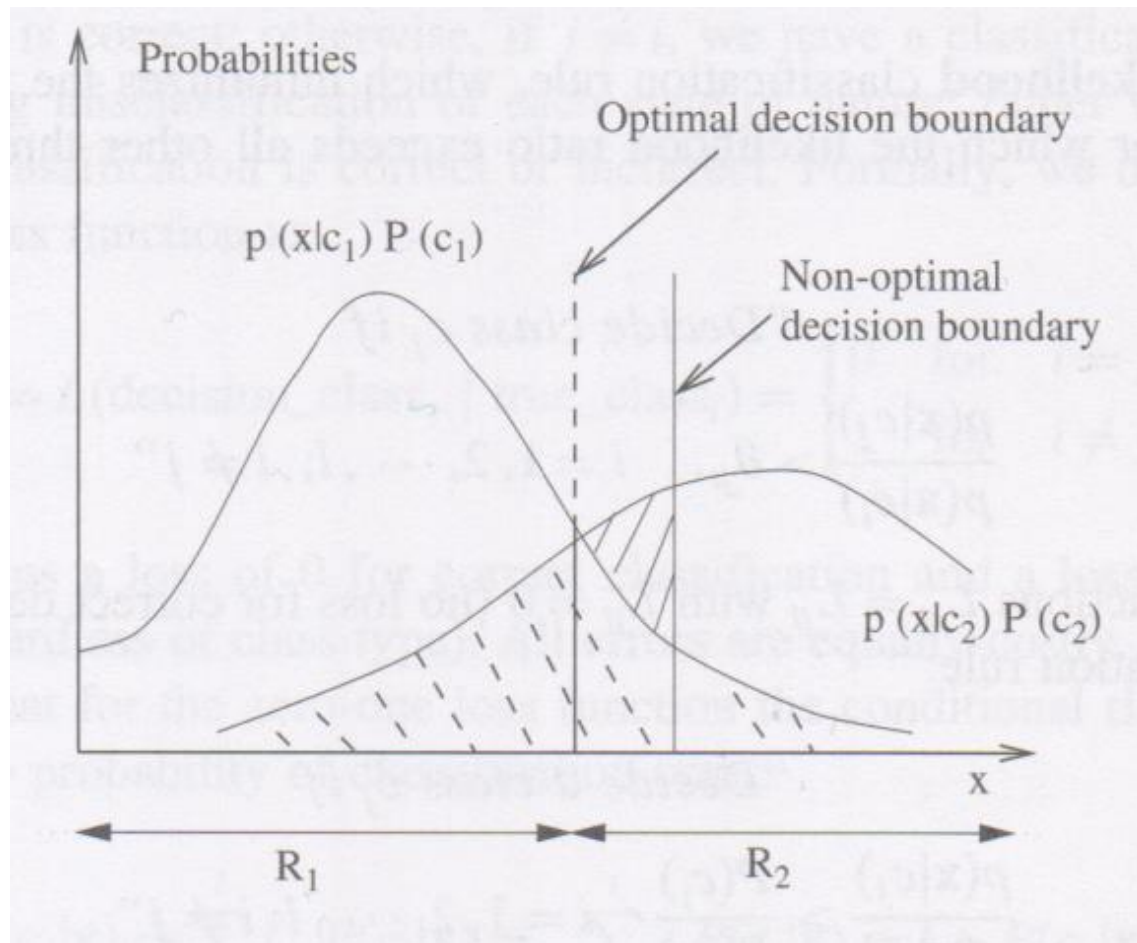


Fig. 3.4 Decision boundaries.

In Fig. 3.5 there is a slightly more complicated classifier.

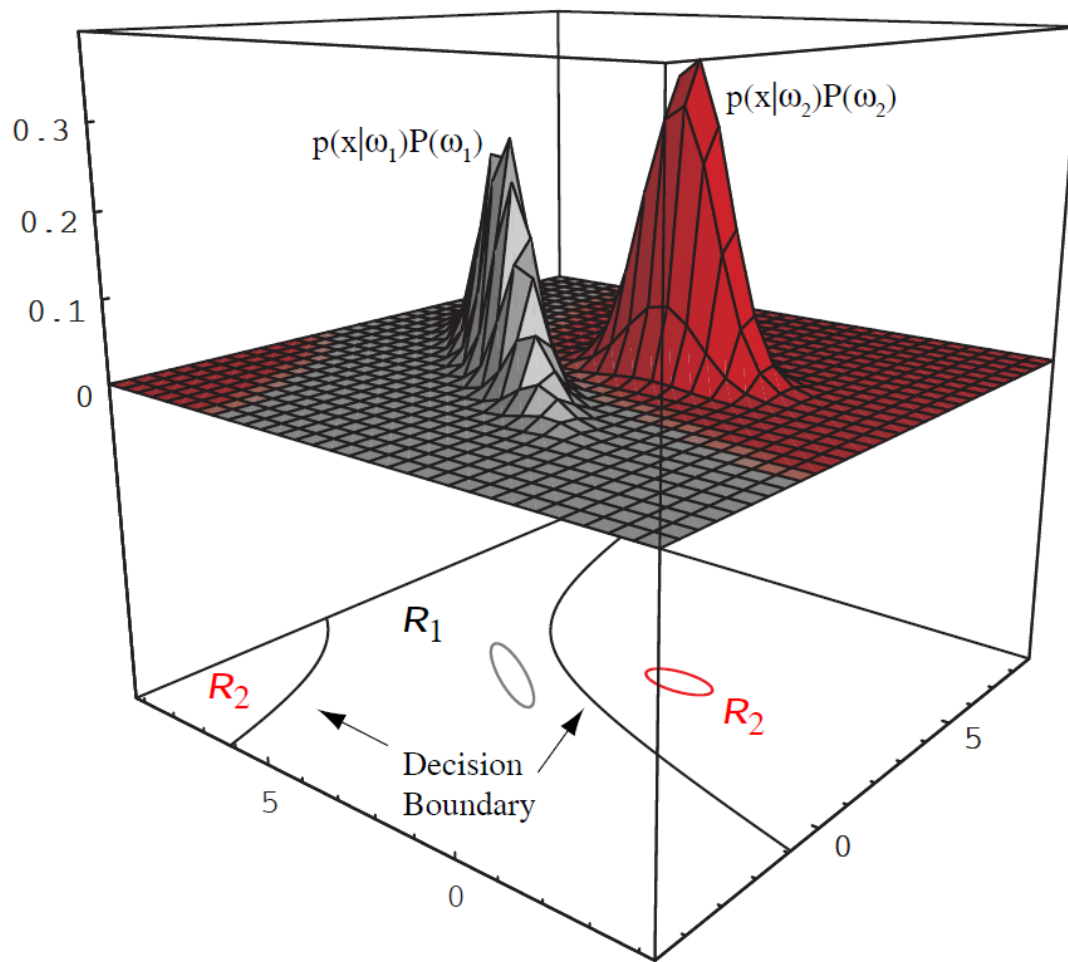


Fig. 3.5 This is a two-dimensional two-class classifier where the probability densities are Gaussian, the decision boundary consists of two hyperbolas, and thus the decision region R_2 is not simply connected. The classes are ω_1 and ω_2 .

3.3 Naive Bayes classifier

Let us return back to the MAP outcome, Equation (3.3) p. 91. It could be computed as described above. However, it is more usual that there are several variable values of the vector. Then we should estimate (the second subscript indexes the vector elements)

$$P(\mathbf{x}_j | c_i) = P(x_{j1}, x_{j2}, \dots, x_{jp} | c_i)$$

by looking at the frequencies (histogram) of all training cases. As the dimensionality of \mathbf{x} increases (the number of variables p gets greater), the amount of data in each interval (bin) of the histogram shrinks. This is the *curse of dimensionality* and means that much more data are needed for training as the dimensionality increases.

There is one simplifying assumption possible to make. We assume that the elements of the variable vector are conditionally independent of each other, given the classification. Given the class c_i , the values of the different variables do not affect each other. This is the "naïveté" in the name of the classifier. It tells that the variables are assumed to be independent of each other. This is a strong assumption and obviously not often true. Nonetheless, we can attempt to model "liberally" real phenomena. This means that the probability of getting the string of variable values

$$P(x_{j1} = a_1, x_{j2} = a_2, \dots, x_{jp} = a_p | c_i)$$

is just equal to the product of multiplying jointly all of the individual probabilities

$$\begin{aligned} &P(x_{j1} = a_1 | c_i) \cdot P(x_{j2} = a_2 | c_i) \cdot \dots \cdot P(x_{jp} = a_p | c_i) \\ &= \prod_k P(x_{jk} = a_k | c_i) \end{aligned}$$

which is much easier to compute and reduces the severity of the curse of dimensionality.

The classification rule is to select class c_i for which the the following is maximum:

$$P(c_i) \prod_k P(x_{jk} = a_k | c_i)$$

This is a great simplification over evaluating the full probability. Interestingly, naive Bayes classifier has been shown to have comparable results to other classification methods for some domains. Where the simplification is true that the variables are conditionally independent of each other, naive Bayes classifier yields exactly the MAP classification.

Example 2

Table 3.2

| Deadline? | Is there a party? | Lazy? | Activity |
|-----------|-------------------|-------|-----------------|
| Urgent | Yes | Yes | Party |
| Urgent | No | Yes | Study |
| Near | Yes | Yes | Party |
| None | Yes | No | Party |
| None | No | Yes | Pub |
| None | Yes | No | Party |
| Near | No | No | Study |
| Near | No | Yes | Computer gaming |
| Near | Yes | Yes | Party |
| Urgent | No | No | Study |

The exemplar training set data in Table 3.2 concerns with what to do in the evening based on whether one has an assignment deadline and what is happening.

We use naive Bayes classifier as follows. We input in the current values for the variables (deadline, whether there is a party, or lazy) and calculate the probabilities of each of four possible things that one might do in the evening based on the data in the training set. Then we select the most likely class.

In reality, probabilities will become very small for the sake of several multiplications with small numbers from interval $[0,1]$. This is a problem with the Bayesian or general probability-based methods.

Let us suppose that one has deadlines looming, but none of them are particularly urgent, i.e., value 'near', that there is 'no party' on, and that one is currently 'lazy'. Then the following classification needs to be evaluated.

- $P(\text{Party}) \cdot P(\text{Near} | \text{Party}) \cdot P(\text{No Party} | \text{Party}) \cdot P(\text{Lazy} | \text{Party})$
- $P(\text{Study}) \cdot P(\text{Near} | \text{Study}) \cdot P(\text{No Party} | \text{Study}) \cdot P(\text{Lazy} | \text{Study})$
- $P(\text{Pub}) \cdot P(\text{Near} | \text{Pub}) \cdot P(\text{No Party} | \text{Pub}) \cdot P(\text{Lazy} | \text{Pub})$
- $P(\text{Gaming}) \cdot P(\text{Near} | \text{Gaming}) \cdot P(\text{No Party} | \text{Gaming}) \cdot P(\text{Lazy} | \text{Gaming})$

Calculating probabilities from Table 3.2 according to the frequencies these evaluate to:

$$P(\text{party} | \text{near (not urgent) deadline, no party, lazy}) = \frac{5}{10} \cdot \frac{2}{5} \cdot \frac{0}{5} \cdot \frac{3}{5} = 0$$

$$P(\text{study} | \text{near (not urgent) deadline, no party, lazy}) = \frac{3}{10} \cdot \frac{1}{3} \cdot \frac{3}{3} \cdot \frac{1}{3} = \frac{1}{30}$$

$$P(\text{pub} | \text{near (not urgent) deadline, no party, lazy}) = \frac{1}{10} \cdot \frac{0}{1} \cdot \frac{1}{1} \cdot \frac{1}{1} = 0$$

$$P(\text{gaming} | \text{near (not urgent) deadline, no party, lazy}) = \frac{1}{10} \cdot \frac{1}{1} \cdot \frac{1}{1} \cdot \frac{1}{1} = \frac{1}{10}$$

Obviously, one will be gaming.

3.4 Some useful statistical formulas

We recall a few statistical concepts useful for the next subsection and later.

Mean, mode and median are known to us. A new concept may be *expectation* or *expected value* that is the probability theory counterpart for the arithmetic mean. Thus, in practice we calculate – for random variable X – the mean taking into account probabilities of different values. For instance, for a dice with probabilities p_j the expected value is calculated:

$$E(X) = \sum_{j=1}^k p_j x_j = \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = 3.5$$

In another example there are 200 000 raffle tickets for 1 €, and one knows that there is one available prize of 100 000 €. Then the expected value of a randomly selected ticket is

$$E = -1 \cdot \frac{199999}{200000} + 99999 \cdot \frac{1}{200000} = -0.5$$

in which the -1 is the price of one's ticket, which does not win 199 999 times out of 200 000 and the 99 999 is the prize minus 1 (the cost on one's ticket). The expected value is not a real value, since one will never actually get 0.5 € back. Thus, the expected value is the mean of the cost and win multiplied by their probabilities according to the formula of the preceding page.


The *variance* of the set of numbers is a measure of how spread out the values are. It is calculated as the sum of the squared distances between each element in the set and the expected value of the set when μ_j is the mean and x_{ij} is a value of variable X_j :

$$\text{var}(\{X_j\}) = \sigma^2(\{X_j\}) = E((\{X_j\} - \mu_j)^2) = \sum_{i=1}^n p_{ij}(x_{ij} - \mu_j)^2$$

The *square root* of variance is known as *standard deviation*, σ_j .

Let us recall the notation for the data matrix or array used for n cases and p variables.

$$\mathbf{D} = \begin{pmatrix} x_{11} & \dots & x_{1j} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nj} & \dots & x_{np} \end{pmatrix}$$

variable
 x_j


The variance looks at the variation in one variable compared to its mean. By generalizing to two variables we get *covariance* which is a measure of how dependent the two variables are in statistical sense:

$$\text{cov}(\{X_j\}, \{X_l\}) = E(\{X_j\} - \mu_j)E(\{X_l\} - \mu_l)$$

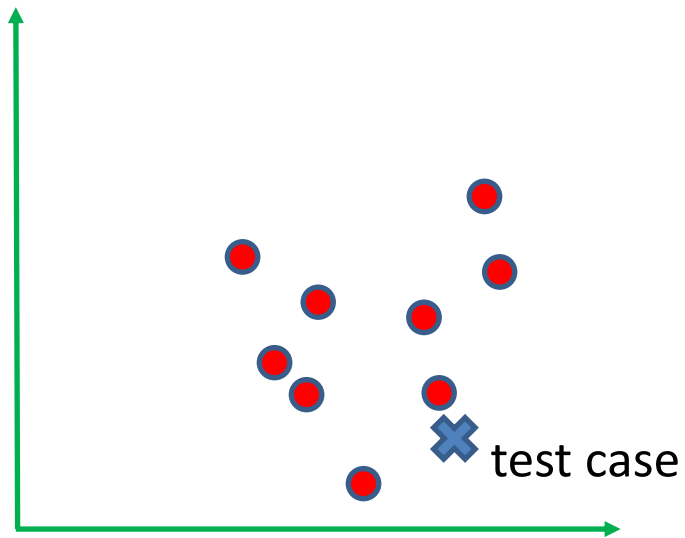
If two variables X_j and X_l are independent, then covariance is 0 (they are uncorrelated), while if both increase and decrease at the same time, covariance is positive, and if one goes up while the other goes down, covariance is negative.

Ultimately, we define the *covariance matrix* that measures correlation between all pairs of variables within a data set:

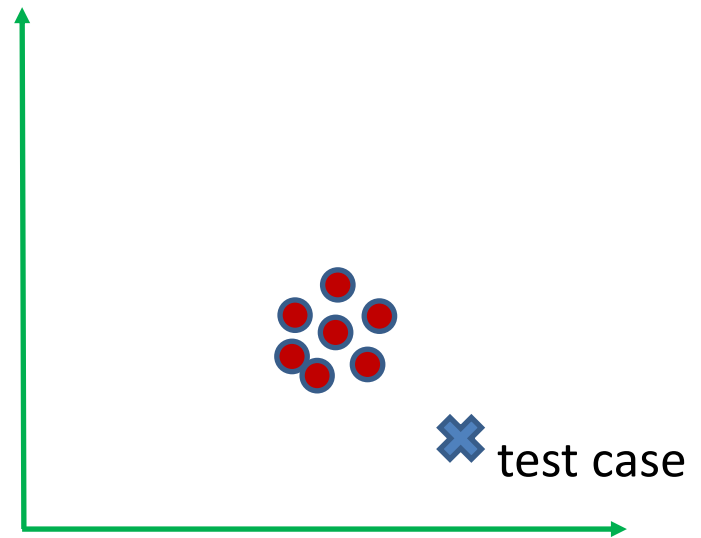
$$\Sigma = \begin{pmatrix} E[(X_1 - \mu_1)(X_1 - \mu_1)] & E[(X_1 - \mu_1)(X_2 - \mu_2)] & \dots & E[(X_1 - \mu_1)(X_p - \mu_p)] \\ E[(X_2 - \mu_2)(X_1 - \mu_1)] & E[(X_2 - \mu_2)(X_2 - \mu_2)] & \dots & E[(X_2 - \mu_2)(X_p - \mu_p)] \\ \dots & \dots & \dots & \dots \\ E[(X_p - \mu_p)(X_1 - \mu_1)] & E[(X_p - \mu_p)(X_2 - \mu_2)] & \dots & E[(X_p - \mu_p)(X_p - \mu_p)] \end{pmatrix}$$

Here X_j is the j th variable describing its values. The covariance is symmetric, $\text{cov}(\{X_j\}, \{X_l\}) = \text{cov}(\{X_l\}, \{X_j\})$.

The covariance matrix presents how the data vary along with each dimension (variable). Fig. 3.5 shows two data sets. The test case is \mathbf{x} . Is it a part of data? For Fig. 3.5 (a) one may answer 'yes', and for (b) 'no', even though the distance from \mathbf{x} to the center is equal on both Fig. 3.5 (a) and (b). The reason is that while looking at the mean, one also looks at where the test case lies in relation to the spread of the actual data points. If the data are tightly controlled, then the test case has to be close to the mean, while if the data are spread out, then the distance of the test case from the mean does not matter as much. Mahalanobis distance takes this into account.



(a)



(b)

Fig. 3.5 Two different data sets and a test case.

Euclidean distance measure can be generalized applying the inverse covariance matrix which yields *Mahalanobis distance*:

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$

where \mathbf{x} is the data arranged as a column vector, $\boldsymbol{\mu}$ is a column vector representing the mean and $\boldsymbol{\Sigma}^{-1}$ is the inverse covariance matrix. If we set identity matrix \mathbf{I} instead, Mahalanobis distance reduces to Euclidean distance.

3.5 The Gaussian

The best known probability distribution is the *Gaussian* or *normal distribution*. In one dimension it has the familiar bell-shaped curve and its equation is

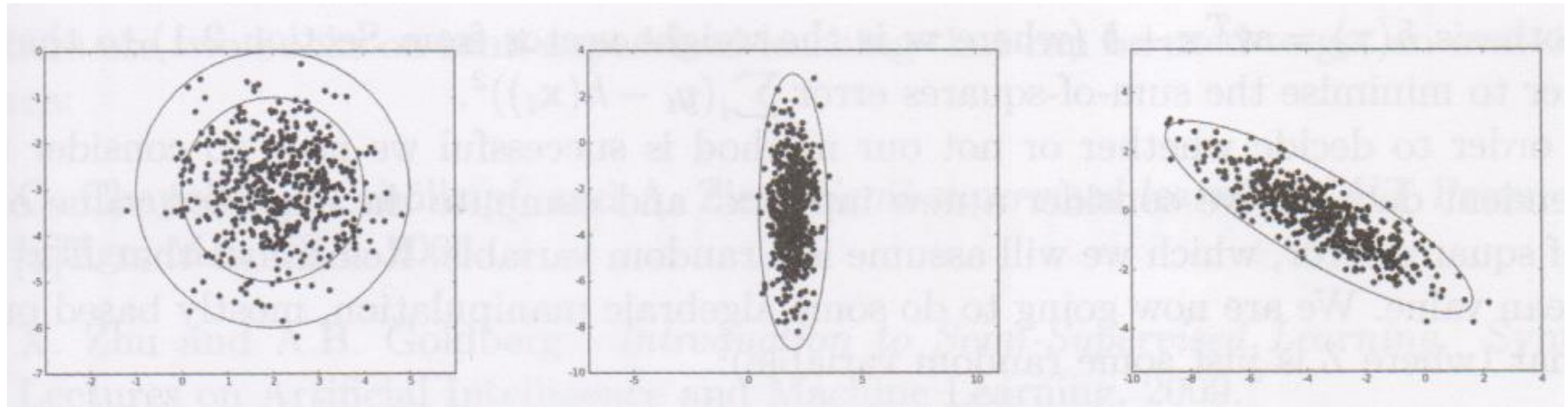
$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

where μ is the mean and σ the standard deviation. In higher dimension p it looks like

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} |\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right) \quad (3.4)$$

where $\mathbf{\Sigma}$ is $p \times p$ covariance matrix and $|\mathbf{\Sigma}|$ its determinant.

Fig. 3.6 shows the appearance in two dimensions of three different cases: (a) when the covariance matrix is the identity, (b) when there are elements only on the leading diagonal of the covariance matrix and (c) the general case.



(a)

(b)

(c)

Fig. 3.6 The two-dimensional Gaussian when (a) the covariance matrix is identity, so-called spherical covariance matrix. (b) and (c) define ellipses, either aligned with the axes (with p parameters) or more generally (with p^2 parameters).

Example 3

Let us look at the the classification task in which there were the crime and demographic data of the UN with 28 variables from 56 countries.⁶ The variables were, for example, adult illiteracy per cent, annual birth rate per 1000 people, life expectancy in years, prisoners per capita, annual software piracies per 100 000 people, and annual murders per 100 000 people.

Classes were constructed as five clusters computed with a self-organizing map (Fig. 3.7). The 56 cases were classified with different classification methods. Here we look at the results of naive Bayes method.

⁶ X. Li, H. Joutsijoki, J. Laurikkala and M. Juhola: Crime vs. demographic factors: application of data mining methods, Webology, Article 132, 12(1), 1-19, 2015.

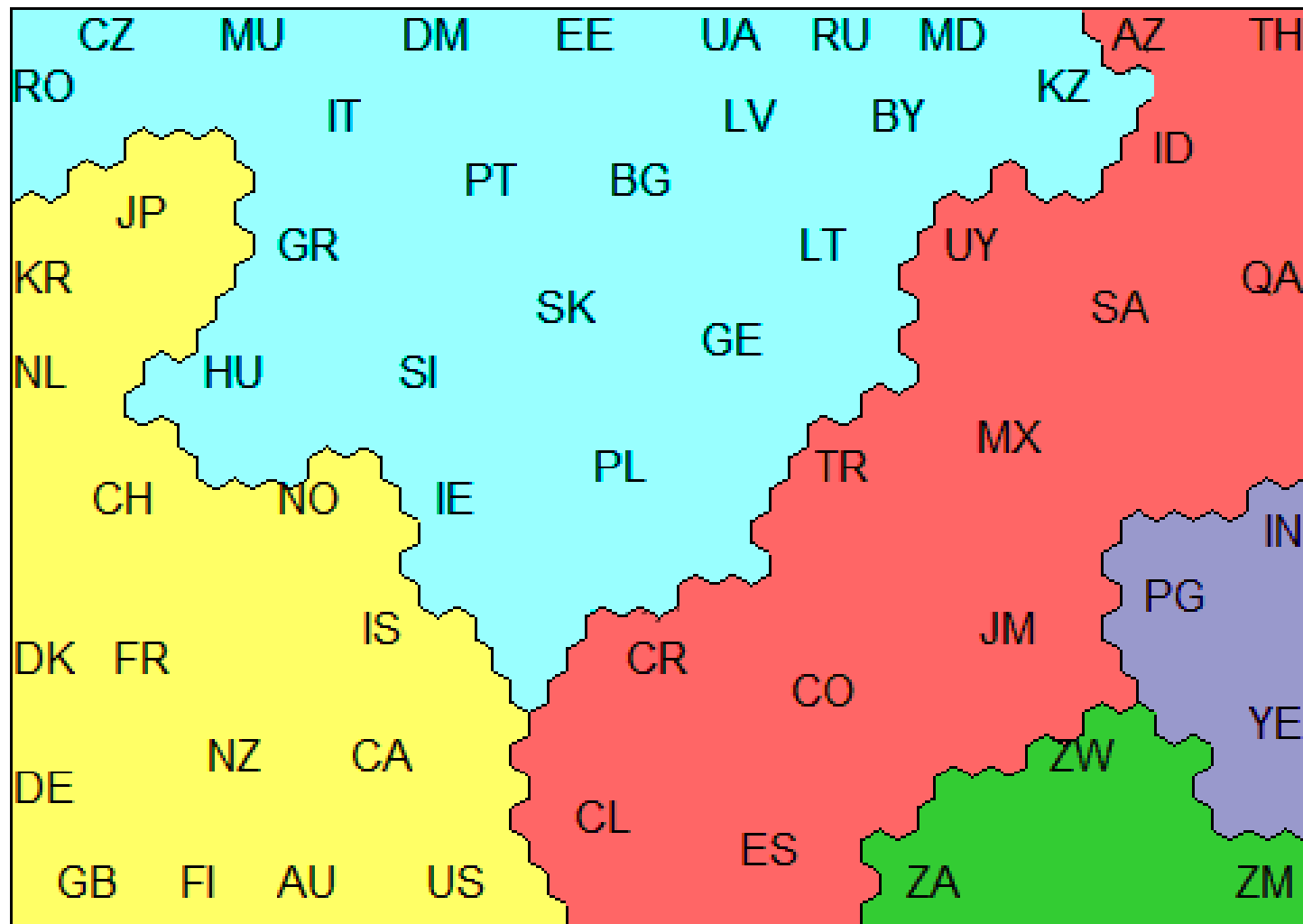


Fig. 3.7 Five clusters of 56 countries computed on the basis of 15 demographic and 13 crime variables.

Table 3.3 Classification accuracies (%) of naive Bayes computed after scaling or without it.

| Method | Not scaled | Scaled into [0,1] | Standardized |
|-------------|------------|-------------------|--------------|
| Naive Bayes | 80.4 | 80.4 | 80.4 |

Before classification no scaling (normalisation), scaling or standardization was run. As we see these do not differ from each other, but all alternatives are shown just to demonstrate their equality and the same manner will later be followed for other methods that may differ. Here there was no difference, because all classification computation was made with probabilities. No scaling, of course, does affect these.

Note that equation (3.4) from p. 117 was not applied when the results in Table 3.3 were computed. These were attained by using the more complicated and time consuming way where for the data of each variable a separate kernel density estimate was computed for each class based on training data of that class. In other words, a kernel distribution was modeled separately for all variables in each class. (This is parameter value 'kernel' for fitcbn of naive Bayes in Matlab.)

Example 4: Vertigo data set

The central variables of Vertigo data set are shown in Table 3.4.⁷ There are 1 nominal variable, 11 binary, 10 ordinal and 16 quantitative variables. The only nominal one was “almost fully ordinal”, because it included four values of which the last one dropping out the first three would form an ordinal variable. We observed that there was only one case with that value from 815 patients. Thus, we applied it “freely” as a single ordinal variable for simplicity. (We could also have left that case out, but not the whole variable that is among the most important.)

⁷ M. Juhola: Data Classification, Encyclopedia of Computer Science and Engineering, ed. B. Wah, John Wiley & Sons, 2008 (print version, 2009, Hoboken, NJ), 759-767.

Table 3.4 Variables and their Types: B = Binary, N = Nominal, O = Ordinal, and Q = Quantitative; Category Numbers After the Nominal and Ordinal

| | | |
|--|---|--|
| [1] patient's age Q | [14] hearing loss type N 4 | [27] caloric asymmetry % Q |
| [2] time from symptoms O 7 | [15] severity of tinnitus O 4 | [28] nystagmus to right Q |
| [3] frequency of spells O 6 | [16] time of first tinnitus O 7 | [29] nystagmus to left Q |
| [4] duration of attack O 6 | [17] ear infection B | [30] pursuit eye movement amplitude gain % Q |
| [5] severity of attack O 5 | [18] ear operation B | [31] and its latency (ms) Q |
| [6] rotational vertigo Q | [19] head or ear trauma: noise injury B | [32] audiometry 500 Hz right ear (dB) Q |
| [7] swinging, floating vertigo or unsteady Q | [20] chronic noise exposure B | [33] audiometry 500 Hz left ear (dB) Q |
| [8] Tumarkin-type drop attacks O 4 | [21] head trauma B | [34] audiometry 2 kHz right (dB) Q |
| [9] positional vertigo Q | [22] ear trauma B | [35] and left ear (dB) Q |
| [10] unsteadiness outside attacks O 4 | [23] spontaneous nystagmus B | [36] nausea or vomiting O 4 |
| [11] duration of hearing symptoms O 7 | [24] swaying velocity of posturography eyes open (cm/s) Q | [37] fluctuation of hearing B |
| [12] hearing loss of right ear between attacks B | [25] swaying velocity of posturography eyes closed (cm/s) Q | [38] lightheadedness B |
| [13] hearing loss of left ear between attacks B | [26] spontaneous nystagmus (eye movement) velocity (°/s) Q | |

Vertigo data set was used for different classification methods, but for Bayes's rule it failed. When computing according to equation (3.4) (p. 117), covariance matrices Σ needed became singular, i.e., not possible to compute their inversions Σ^{-1} . The reason was that the data set includes variables with purely zeros for some classes (not for the whole data). Thus, it was not possible to compute classification results under the given specifications.