Chapter 7
Biclustering

7.1 Clustering in two dimensions

Clustering techniques aim at partitioning a given set of data into clusters. Chapter 3 presents the basic k-means approach and many variants to the standard algorithm. All these algorithms search for an optimal partition in clusters of a given set of samples. The number of clusters is usually denoted by the symbol k. As previously discussed in Chapter 3, each cluster is usually labeled with an integer number ranging from 0 to $k-1$. Once a partition is available for a certain set of samples, the samples can then be sorted by the label of the corresponding cluster in the partition. If a color is then assigned to the label, a graphic visualization of the partition in clusters is obtained. This kind of graphic representation is used often in two-dimensional spaces for representing partitions found with biclustering methods.

A set of data can be represented through a matrix. The samples can be represented by m-dimensional vectors, where the components of these vectors represent the features used for describing each sample. All the vectors representing the samples can be grouped in a matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{pmatrix}.$$

If a given set of data contains n samples which are represented by m features, then A is an $m \times n$ matrix. Each column of the matrix represents one sample, and it provides information on the expression of its m features. Each row represents a feature, and it provides the expression of that feature on the n samples of the set of data.

Standard clustering methods partition the samples in clusters, i.e., the columns of the matrix A are partitioned in clusters. Biclustering methods work instead simultaneously on the columns and the rows of the matrix A. Besides clustering the samples, even their features are partitioned in clusters. Two different partitions are therefore

A. Mucherino et al., *Data Mining in Agriculture*, Springer Optimization and Its Applications 34, 143
DOI: 10.1007/978-0-387-88615-2_7,

needed. The search of the two partitions is not performed independently, but rather the clusters of samples and the clusters of features are related. The concept of "bicluster" is introduced for this purpose. A *bicluster* is a collection of pairs of samples and features subsets $B = \{(S_1, F_1), (S_2, F_2), \ldots, (S_k, F_k)\}$, where k, as usual, is the number of biclusters [32]. Each bicluster (S_r, F_r) is formed by two single clusters: S_r is a cluster of samples, and F_r is a cluster of features.

The following conditions must be satisfied:

$$\bigcup_{r=1}^{k} S_r \equiv A, \qquad S_\zeta \cap S_\xi = \emptyset \quad 1 \leq \zeta \neq \xi \leq k,$$

$$\bigcup_{r=1}^{k} F_r \equiv A, \qquad F_\zeta \cap F_\xi = \emptyset \quad 1 \leq \zeta \neq \xi \leq k.$$

Note that the union of all the clusters S_r must be A because each sample, organized in columns in the matrix, must be contained in at least one cluster S_r. Similarly, the union of all the clusters F_r must be A as well. The only difference is that the features are organized on the rows of the matrix A. Note also that these same conditions are imposed on clusters when standard clustering is applied. Besides ensuring that each single sample or feature is contained in a cluster, they guarantee that all the clusters of samples and the clusters of features are disjoint.

The aim of biclustering techniques is to find a partition of the samples and of their features in biclusters (S_r, F_r). In this way, not only a partition of samples is obtained, but also the features causing this partition are identified. As for the standard clustering, the single clusters S_r and F_r can be labeled from 0 to $k-1$. Independently, the clusters S_r can be sorted by their own labels, and the same can be done for the clusters F_r. A color or a gray scale can be associated to each label, and a matrix of pixels can be created. On the rows of such matrix, the clusters F_r are ordered by their labels, and the clusters S_r are ordered on the columns. Even though this matrix is built considering the clusters S_r and F_r independently, it gives a graphic visualization of the biclusters (S_r, F_r). The matrix shows a checkerboard pattern where the biclusters can be easily identified. This pattern can be easily noticed, for instance, in Figure 7.4, related to the application of biclustering discussed in Section 7.4.1.

Biclustering is widely applied for partitioning gene expression data, and therefore some of the nomenclature in biclustering is similar to the one in gene expression analysis. In [159], a survey of biclustering algorithms for biological data is presented. Since biology is currently the main field of application of biclustering, this survey can be actually considered a survey on biclustering. It is updated to the year 2004, and hence it does not include recent developments, which are discussed in Section 7.2 of this chapter.

Following the definition, a bicluster is a pair of clusters (S_r, F_r), where S_r is a cluster of samples and F_r is a cluster of features. Since the samples and the features are organized in the matrix A as explained above, a bicluster can also be seen as a

submatrix of A. A submatrix of an $m \times n$ matrix can be identified by the set of row indices and column indices it takes from A. For instance, if

$$A = \begin{pmatrix} 1 & 2\ 3 \\ 1 & 1\ 0 \\ 0 & -1\ 2 \end{pmatrix},$$

then the submatrix with the first and third row of A and the second and third column of A is

$$S_A = \begin{pmatrix} 2\ 3 \\ -1\ 2 \end{pmatrix}.$$

In the following, bicluster and submatrix of A will be used interchangeably.

Different kinds of biclusters can be defined. One might be interested in biclusters in which the corresponding submatrices of A have constant values. This requirement may be too strong in some cases, and it may work on non-noisy data only. Indeed, data from real-life applications are usually affected by errors, and a bicluster with constant values may not be possible to find. Formally, these kinds of biclusters are the ones in which

$$a_{ij} = \mu \quad \forall i, j: \quad a_i \in F_r \quad a^j \in S_r,$$

where μ is a real constant value. If the data contain errors, the following formalism can be used

$$a_{ij} = \mu + \eta_{ij} \quad \forall i, j: \quad a_i \in F_r \quad a^j \in S_r,$$

where η_{ij} is the noise associated to a real value μ of a_{ij}. The problem of finding biclusters with constant values can be formulated as an optimization problem in which the variance of the elements of the biclusters have to be minimized. If I_{S_r} is the set of column indices related to the samples $a^j \in S_r$, i.e., I_{S_r} contains all the j indices associated to S_r, and I_{F_r} is the set of row indices related to the features $a_i \in F_r$, then

$$f(S_r, F_r) = \sum_{i \in I_{S_r}} \sum_{j \in I_{F_r}} \left(a_{ij} - M\right)^2$$

evaluates the quality of the bicluster (S_r, F_r), where M is the average of all the elements in (S_r, F_r). If the data are not affected by errors, a *perfect* bicluster with constant values is such that $f(S_r, F_r) = 0$. Otherwise, minimizing the function $f(S_r, F_r)$ equals finding the bicluster which is closest to the optimal one. It is worth noting that every bicluster containing one row and one column is a perfect bicluster with constant values, since its only element a_{ij} equals M. In general, when the function $f(S_r, F_r)$ is optimized, constraints must take into account that the number of rows and columns of the submatrices representing the biclusters must be greater than a certain threshold.

Biclusters with constant row values and constant column values can also be of interest. If the row values in a bicluster are constant, then all the samples in the bicluster (and in S_r) have a constant subset of features (the ones in F_r). Inversely,

if the columns have constant values, then the samples in S_r have all the features in F_r constant. In this case, different samples have different feature values, but all the feature values in the same sample are the same. A bicluster having constant rows satisfies the condition

$$a_{ij} = \mu + \alpha_i \quad \forall i, j : \quad a_i \in F_r \quad a^j \in S_r$$

or the condition

$$a_{ij} = \mu\alpha_i \quad \forall i, j : \quad a_i \in F_r \quad a^j \in S_r$$

where μ is a typical value within the bicluster and α_i is the adjustment for row $i \in I_{S_r}$. Similarly, a bicluster having constant columns satisfies the condition

$$a_{ij} = \mu + \beta_j \quad \forall i, j : \quad a_i \in F_r \quad a^j \in S_r$$

or the condition

$$a_{ij} = \mu\beta_j \quad \forall i, j : \quad a_i \in F_r \quad a^j \in S_r.$$

Even here, the presented conditions can be satisfied only if the data are not noisy, otherwise the noise parameters η_{ij} can be used, as in the previous example of biclusters with constant values.

The easiest way to approach the problem of finding biclusters with constant row values or constant column values is the following one. Let us suppose a bicluster with constant rows is contained in a matrix A and that the submatrix which corresponds to it is

$$S_A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}.$$

Since all the values on the rows are constant, the mean among all these values corresponds to any of the row values. If each row is normalized by the mean of all its values, then the following matrix is obtained

$$\hat{S}_A = \begin{pmatrix} 1/1 & 1/1 & 1/1 & 1/1 \\ 2/2 & 2/2 & 2/2 & 2/2 \\ 3/3 & 3/3 & 3/3 & 3/3 \\ 4/4 & 4/4 & 4/4 & 4/4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

which corresponds to a bicluster with constant values. Therefore, the row and the columns normalization can allow the identification of biclusters with constant values on the rows or on the columns of the matrix A by transforming these biclusters into constant biclusters.

Biclusters have coherent values when the generic element of the corresponding submatrix can be written as

$$a_{ij} = \mu + \alpha_i + \beta_j \quad \forall i, j : \quad a_i \in F_r \quad a^j \in S_r.$$

Particular cases of coherent biclusters are biclusters with constant rows ($\beta_j = 0$), or biclusters with constant columns ($\alpha_i = 0$), or biclusters with constant values ($\alpha_i = \beta_j = 0$). This kind of bicluster can be represented by submatrices such as

$$S_A = \begin{pmatrix} \mu + \alpha_1 + \beta_1 & \mu + \alpha_1 + \beta_2 & \dots & \mu + \alpha_1 + \beta_m \\ \mu + \alpha_2 + \beta_1 & \mu + \alpha_2 + \beta_2 & \dots & \mu + \alpha_2 + \beta_m \\ \dots & \dots & \dots & \dots \\ \mu + \alpha_n + \beta_1 & \mu + \alpha_n + \beta_2 & \dots & \mu + \alpha_n + \beta_m \end{pmatrix}.$$

The whole submatrix S_A can be built using the value μ and the two vectors $\alpha \equiv (\alpha_1, \alpha_2, \dots, \alpha_n)$ and $\beta \equiv (\beta_1, \beta_2, \dots, \beta_m)$.

The following proves that a generic element a_{ij} of a submatrix S_A can be obtained from means among the rows, the columns and all the elements of the matrix. The mean among the elements of the i^{th} row of S_A is

$$M_i = \mu + \alpha_i + \frac{1}{m} \sum_{k=1}^{m} \beta_k,$$

whereas the mean among the elements of the j^{th} column of S_A is

$$M_j = \mu + \frac{1}{n} \sum_{k=1}^{n} \alpha_k + \beta_j.$$

Moreover, the mean of all the elements of the matrix S_A is

$$M = \mu + \frac{1}{n} \sum_{k=1}^{n} \alpha_k + \frac{1}{m} \sum_{k=1}^{m} \beta_k.$$

From simple computations, it results that

$$M_i + M_j - M = \mu + \alpha_i + \beta_j = a_{ij}. \tag{7.1}$$

Therefore, the generic element of a coherent bicluster can be written as the mean of its rows, plus the mean of its columns, minus the mean of the whole submatrix. If the data are affected by errors, then equation (7.1) may not be satisfied. The residue $r(a_{ij})$ associated to an element a_{ij} is then defined as

$$r(a_{ij}) = a_{ij} - M_i - M_j + M$$

and consists of the difference between the value a_{ij} and the value obtained applying equation (7.1). A perfect (not affected by noise) coherent bicluster would have all the residues $r(a_{ij})$ equal to zero. Thus, the following function is able to evaluate the coherency of biclusters:

$$H(S_r, F_r) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left[r(a_{ij}) \right]^2 .$$

Coherent biclusters can be located in the matrix A by minimizing this objective function.

As shown in this section, the problem of finding a bicluster or a partition in biclusters can be formulated as an optimization problem. The easiest way to solve it is through an exhaustive search among all the possible biclusters. This can be affordable only if the considered set of data contains a small number of samples and features. When this is not the case, optimization methods need to be used. In Section 1.4, some standard methods for optimization are presented. However, usually the optimization methods used for biclustering are tailored to the particular problem to solve [66, 83].

7.2 Consistent biclustering

In this section, the notion of consistent biclustering is introduced. This part of the chapter makes a large use of mathematical symbols: the symbology utilized follows. As already observed, the set of clusters S_r and the set of clusters F_r represent two partitions of the samples and of the features of a set of data. Each cluster S_r or F_r has a certain center. Since we have to deal with two different partitions (samples and features), let us denote the center of the generic cluster S_r with the symbol c_r^S and the center of the generic cluster F_r with the symbol c_r^F. The center c_r^S refers to the r^{th} cluster of the samples. Since it is the average of samples represented by m-dimensional vectors, c_r^S is an m-dimensional vector. These vectors can be organized into an $m \times k$ matrix C_S, where the centers are stored column by column, just as the samples in the matrix A. The same can be done in correspondence of the clusters F_r and their centers. The generic center c_r^F refers to the r^{th} cluster of features. A matrix C_F can be defined where such centers are organized column by column. C_F is an $n \times k$ matrix, since each feature is represented by an n-dimensional vector. Since the matrices C_S and C_F contain averages, their elements are the average expressions of the corresponding samples and features. It is clear that the nomenclature "average expression'' comes from the studies on gene expression data. An average expression can be evaluated by a non-negative number: we will suppose in the following that all the centers have non-negative values.

Matrices are widely used in biclustering: A contains the set of data to partition in biclusters; C_S and C_F contain the centers of the clusters S_r and F_r, respectively. a_{ij} refers to the i^{th} feature of the j^{th} sample. A sample can be referred to as a^j: the j as superscript means that the j refers to the column index of the matrix A. Similarly, a_i refers to the i^{th} row of the matrix, i.e., to the i^{th} feature. The same symbology can be used for elements in C_S and C_F. c_{ir}^S refers to the i^{th} component of the center of the cluster S_r; c_{jr}^F refers to the j^{th} component of the center of the cluster F_r.

As already pointed out, the two single clusters in a bicluster (S_r, F_r) are related. Actually, once a partition in clusters of the samples is provided, a corresponding partition in clusters of the features can be obtained. Vice versa, a partition in clusters S_r can be obtained from the clusters F_r. Let us suppose then that the clusters S_r are known. In this case, each sample or column a^j is assigned to a certain cluster. The centers of all the clusters S_r are also known and contained in the matrix C_S column by column. The generic element c_{ir}^S of the matrix represents the average expression of the i^{th} feature in the r^{th} cluster, among all the samples in S_r. Let $\hat{r}$ be the cluster in which the i^{th} feature is most expressed. In mathematical formulas, $\hat{r}$ can be defined as the index such that the following condition is satisfied:

$$a_i \in F_{\hat{r}} \iff c_{i\hat{r}}^S > c_{i\xi}^S \quad \forall \xi \in \{1, 2, \ldots, k\} \quad \xi \neq \hat{r}. \tag{7.2}$$

Intuitively, it is reasonable to assign the feature a_i to the cluster $F_{\hat{r}}$. If the condition (7.2) is applied for all the indices $i \in \{1, 2, \ldots, k\}$ and all the features a_i are assigned to the corresponding clusters $F_{\hat{r}}$, a partition in clusters F_r is obtained from a previous partition in clusters S_r.

The same procedure can be applied for obtaining a partition of the samples when a partition of the features is known. The following rule can be used for assigning a sample a^j to a certain cluster $\hat{S}_r$:

$$a^j \in \hat{S}_{\hat{r}} \iff c_{j\hat{r}}^F > c_{j\xi}^F \quad \forall \xi \in \{1, 2, \ldots, k\} \quad \xi \neq \hat{r}. \tag{7.3}$$

If this rule is applied for each j, a new partition in clusters $\hat{S}_r$ is obtained from the partition in clusters F_r. Note that a symbol is used for discriminating the generic cluster S_r and the generic cluster $\hat{S}_r$. Indeed, S_r is the generic cluster used for finding a partition in clusters F_r of the features, whereas $\hat{S}_r$ represents the partition in clusters obtained from the clusters F_r. Two different notations for S_r and $\hat{S}_r$ are used because these two partitions of samples can be different in general. Even though S_r generated F_r and F_r generated $\hat{S}_r$, there are no reasons why S_r and $\hat{S}_r$ should correspond. If they correspond, then the partition in biclusters (S_r, F_r) is called *consistent*.

It is important to note that not all the sets of data admit a consistent partition in biclusters. This may happen because there may not be a statistical evidence that a sample or a feature belongs to a certain cluster. If a consistent partition in biclusters exists for a certain set of data, then it is said to be *biclustering-admitting*. When it is not the case, samples or features are usually deleted from the set of data for letting it become biclustering-admitting. In this case, it is important to delete the least possible in order to preserve the information in the set of data. This procedure is known as *feature selection*.

The requirement of consistency can be weak in some cases. Let us suppose that a partition in clusters S_r is available, and that a partition in clusters F_r is obtained from it. Each feature is therefore assigned to the cluster $F_{\hat{r}}$ such that $c_{i\hat{r}}^S$ has the largest value in the vector c_i^S. Let us suppose now that the following condition holds:

$$\min_{\xi \neq \hat{r}} \{c_{i\hat{r}}^S - c_{i\xi}^S\} \leq \varepsilon \tag{7.4}$$

where ε is a small number. In this case, small changes in the data can bring different partitions of the features in the clusters F_r. Indeed, small variations of the samples bring variations of the centers of the clusters S_r, and this can bring a different feature to be more expressed. The following example should clarify this concept.

Let us suppose that the data are partitioned in two biclusters only. S_1 and S_2 are known, as well as their centers c_1^S and c_2^S. The features are also partitioned into two clusters F_1 and F_2. Each feature is assigned to one of the two clusters depending on their average expressions in the corresponding clusters S_r. Therefore, the generic feature a_i is assigned to F_1 if $c_{i1}^S > c_{i2}^S$, and vice versa. Let us suppose for instance that $c_{i1}^S = 5.9$ and $c_{i2}^S = 6.1$. Then, a_i is assigned to F_2. However, the condition (7.4) holds with $\alpha \geq 0$. This means that it is not evident statistically that a_i belongs to F_2. Indeed, let us suppose that another sample is added to the set of data, and that it is assigned to cluster S_1. The center of S_1 hence changes, and in particular its i^{th} component changes. If the feature a_i is more expressed in this sample, the average c_{i1}^S can increase. Since it is an average and it considers all the samples in the same cluster, it cannot change dramatically, even though the new sample might be different from the others. However, in the considered example, the feature a_i might be assigned to a different cluster after the new sample is added. If indeed c_{i1}^S is now equal to 6.2, then $c_{i1}^S > c_{i2}^S$, and the feature a_i is assigned to F_1.

In order to overcome this kind of problem, conditions stronger than consistent biclustering are introduced in [176]. A biclustering is called an *additive consistent biclustering* with parameter α or an *α-consistent biclustering* if the following two relations holds

$$a_i \in \hat{F}_{\hat{r}} \iff c_{i\hat{r}}^S > \alpha_j^F + c_{i\xi}^S \quad \forall \xi \in \{1, 2, \ldots, k\} \quad \xi \neq \hat{r} \tag{7.5}$$

$$a^j \in \hat{S}_{\hat{r}} \iff c_{j\hat{r}}^F > \alpha_i^S + c_{j\xi}^F \quad \forall \xi \in \{1, 2, \ldots, k\} \quad \xi \neq \hat{r} \tag{7.6}$$

where each α_j^F and α_i^S are positive numbers. It is easy to prove that an α-consistent biclustering is a consistent biclustering, but not the inverse. Indeed, if the conditions (7.5) and (7.6) are satisfied with $\alpha_j^F > 0$ and $\alpha_i^S > 0$, then they keep being satisfied with $\alpha_j^F = 0$ and $\alpha_i^S = 0$. Inversely, let us suppose that $c_{i\hat{r}}^S > \alpha_j^F + c_{i\xi}^S$ for all the ξ different from $\hat{r}$, in correspondence with some feature a_i and with $\alpha_j^F = 0$. If α_j^F is successively modified and it becomes positive, then the condition may not be satisfied anymore. The quantity $\alpha_j^F + c_{i\xi}^S$ becomes larger, and therefore the quantity $c_{i\hat{r}}^S$ may not be greater than it anymore.

Similar to α-consistent biclustering is the β-consistent biclustering. A biclustering is called a *multiplicative consistent biclustering* with parameter β or a *β-consistent biclustering* if the following two relations holds

$$a_i \in \hat{F}_{\hat{r}} \iff c_{i\hat{r}}^S > \beta_j^F c_{i\xi}^S \quad \forall \xi \in \{1, 2, \ldots, k\} \quad \xi \neq \hat{r} \tag{7.7}$$

$$a^j \in \hat{S}_{\hat{r}} \iff c_{j\hat{r}}^F > \beta_i^S c_{j\xi}^F \quad \forall \xi \in \{1, 2, \ldots, k\} \quad \xi \neq \hat{r} \tag{7.8}$$

where $\beta_j^F > 1$ and $\beta_i^S > 1$. As before, a β-consistent biclustering is a consistent biclustering.

7.3 Unsupervised and supervised biclustering

Biclustering is a technique for clustering on two dimensions. On the first dimension, the samples contained in a set of data are taken into account. Standard clustering methods work on this dimension only. On the second dimension, moreover, biclustering considers the features that are used for representing the samples. The simultaneous clustering of samples and features allows one to partition the data in clusters where similar samples are contained, and to find out the features that cause these similarities.

Biclustering can be performed by solving one of the optimization problems discussed in Section 1.4. In this way, the partition of the samples and the partition of the features are searched simultaneously. Biclustering can also be performed by using methods for standard clustering coupled with the concepts introduced in the previous section. For instance, the k-means algorithm can be applied for partitioning a given set of samples. Then, the conditions (7.2) can be used for finding a correspondent partition in clusters of the features. In this way, the biclusters can be defined. Besides the partition of the samples, the partition of their features allows one to identify the ones that generate the current partition of the samples.

However, the partition found in biclusters might not be consistent. From the partition in clusters of the features, a partition in clusters, the samples can be obtained using the conditions (7.3). As already pointed out, the obtained partition of the samples can be equal or not to the starting partition, i.e., to the partition found by the k-means algorithm in this example. If they correspond, the biclustering is consistent, otherwise it is not. In the latter case, some features be can deleted from the set of data in order to let the biclustering become consistent. The feature selection process is not easy, and the consistent biclustering can be found only if the set of data is biclustering-admitting.

Clustering techniques are referred to as techniques for unsupervised classifications, because they are used when there is not any previous knowledge about the data. Biclustering can be also supervised, because the information from a training set can be actually used. If a training set is available, a set of data is available that is already partitioned in different classes. In this case, a partition algorithm such as k-means is not needed, because the data are already partitioned. Then, a partition of the features can be obtained applying the conditions (7.2). At this point, a set of biclusters is defined, which is able to provide information on the features that caused the classification of the samples given by the training set. As before, this information is accurate if the biclustering is consistent, otherwise there is not a strong statistical evidence that a feature belongs to one cluster or another.

The problem of finding a consistent biclustering, once a partition of the samples is given, can be formulated as an optimization problem (see Section 1.4). Before

formulating the optimization problem, let us introduce some notations. Let F be an $m \times k$ matrix whose elements can have value 0 or 1 only. The generic f_{ir} element has value 1 if the feature a_i belongs to the cluster F_r, and 0 otherwise. By using this matrix, the condition of consistency can be written as follows. Suppose that the clusters S_r are known. Suppose that the clusters F_r are built by using the conditions (7.2). Then, the clustering in biclusters (S_r, F_r) is consistent if S_r is obtained when the conditions (7.3) are applied. Equivalently, the following conditions must hold:

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}}}{\sum_{i=1}^{m} f_{i\hat{r}}} > \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi}}{\sum_{i=1}^{m} f_{i\xi}}, \quad \forall \hat{r}, \xi \in \{1, 2, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}}. \tag{7.9}$$

Let us introduce now the binary vector x of length m whose generic element x_i is 1 if the feature a_i is taken into account, and 0 otherwise. The condition (7.9) on a subset of features can be written as follows:

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, 2, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}}. \tag{7.10}$$

As already pointed out, when deleting features in order to find a consistent biclustering, the minimum possible features have to be removed. The problem of choosing a subset of features that is as large as possible and such that the corresponding biclustering is consistent can be formulated as an optimization problem. The function to maximize is

$$f(x) = \sum_{i=1}^{m} x_i \tag{7.11}$$

while subject to the constraints (7.10). In the optimization field, this problem is called *fractional 0-1 programming problem*. Its solution provides an efficient selection of the features to take into account. This optimization problem can be solved by using a suitable method for global optimization (Section 1.4), but it is usually quite difficult to manage. Therefore, ad hoc methods have been developed. Details about these methods can be found in [32, 176].

The solutions of the formulated optimization problem allow one to obtain consistent biclusterings where the maximum number of features is considered. Similarly, the following optimization problem provides α-consistent biclusterings:

$$\max_x f(x)$$

subject to

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \alpha_j + \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, 2, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}}.$$

This other optimization problem provides instead β-consistent biclusterings:

$$\max_x f(x)$$

subject to

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \beta_j \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, 2, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}}.$$

7.4 Applications

Biclustering techniques are nowadays mainly applied to the field of biology, and in particular for the analysis of microarray data. In Section 7.4.1 we will discuss in detail this kind of application and we will report the experiments presented in [32], where supervised biclustering has been applied. Moreover, even other applications of biclustering have emerged in the literature. Biclustering is used for collaborative filtering, where the aim is to identify subgroups of customers with similar preferences or behaviors toward a subset of products [55, 228, 244]. In information retrieval and text mining [60], biclustering can be successfully used to identify subgroups of documents with similar properties relative to subgroups of attributes, such as words or images. In [103], biclustering has been used for analyzing electoral data and, in [142], it has been used for studying the exchanges of foreign currencies. To the best of our knowledge, biclustering has never been used before for solving problems related to agriculture. However, as we will explain in Section 7.4.2, it is our opinion that biclustering techniques can be successfully applied to agricultural-related data mining problems.

7.4.1 Biclustering microarray data

Microarrays in biology are used for studying the expression of genes under different conditions. Genes in humans, for instance, have different expression levels in presence of diseases. Finding the set of genes that have similar expression levels in the

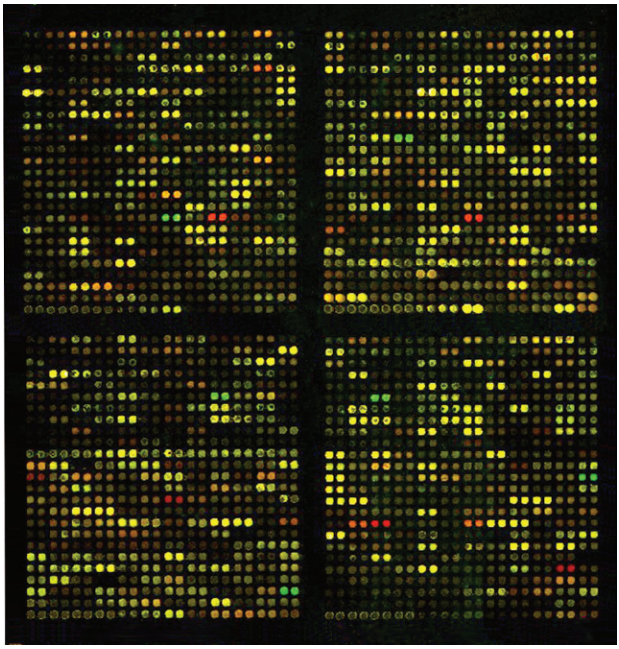

Fig. 7.1 A microarray.

presence of a certain disease can help understanding the disease itself and how the body reacts to it. Microarray data are organized as in a matrix: each row of the matrix is related to a gene, and each column is related to a different condition. Therefore, the generic element of a microarray gives the expression level of the gene, specified by the current row, under the condition specified by the current column. The expression levels are usually visualized by a matrix of colors ranging from light green to red. In black and white pictures, this range of colors corresponds to a gray scale from white to black. Figure 7.1 shows a microarray.

The expression levels obtained by a microarray can be placed in a $m \times n$ numerical matrix A. The samples contained in this matrix are organized column by column: each of them represents an experimental condition through the expression levels of all the considered genes. The features used for describing such samples are hence the expression levels of the genes. Each row of A contains all the measured expression levels of the same gene under the different experimental conditions.

Biclusters in the matrix A can reveal genetic pathways that can be used, for instance, for identifying the genes with different expression levels in presence of a disease. A bicluster of samples and features groups a subset of similar conditions that are caused by a subset of genes having similar expression levels. The meaning of the term "similar" depends on the kind of considered bicluster. For instance, biclusters can have constant values, on the whole bicluster or only on its rows or columns, or it can be a bicluster with coherent values.

Another way for finding biclusters in the matrix A is to look for a consistent biclustering of the data as explained in Section 7.2. Let us suppose that the samples (the experimental conditions in this application) are already classified in clusters. Then, the rule (7.2) can be used for finding a partition in clusters of the features, i.e., a partition in clusters of the genes. In this way, biclusters containing conditions and genes can be identified, and the genes causing certain conditions can be located. It is important to note that the correlation between conditions and genes is statistically evident only if the partition found in biclusters is consistent. For this reason, the best way to find such partition is to solve the optimization problem (7.11)–(7.10). In this way, the features that cause the biclustering not to be consistent are removed.

In [32, 176], this technique has been applied to a well-researched microarray data set containing samples from patients diagnosed with *acute lymphoblastic leukemia* (ALL) and *acute myeloid leukemia* (AML) diseases [89]. The original set of data has been divided in two parts: a part used as training set and another used as validation set. Hence, the training set used contains 27 samples classified as ALL and 11 sample classified as AML; the validation set contains 20 ALL samples and 14 AML samples. A consistent biclustering is obtained by following a methodology described in [32], which is based on the optimization of the problem (7.11)–(7.10). After that, the samples of the validation set are subsequently classified choosing for each of them the class with the highest average feature expression: 3439 features for class ALL and 3242 features for class AML have been selected. The obtained classification contains only one error: one AML-sample was classified into the ALL class. The obtained partition in biclusters is shown in Figure 7.2.

The same methodology has also been applied to the *Human Gene Expression* (HuGE) Index data set [112]. The purpose of the HuGE project is to provide a comprehensive database of gene expressions in normal tissues of different parts of the human body and to highlight similarities and differences among the organ systems [111]. The data set consists of 59 samples from 19 distinct tissue types. It was obtained using oligonucleotide microarrays capturing 7070 genes. The samples were obtained from 49 human individuals: 24 males with median age of 63 and 25 females with median age of 50. Each sample came from a different individual except for the first 7 BRA (brain) samples that were from different brain regions of the same individual and 5th LI (liver) sample, which came from that individual as well. The list of considered tissue types with their abbreviations and the number of samples for each of them is given in Figure 7.3. Figure 7.4 presents the partition in biclusters obtained by applying the same methodology as above. The distinct block-diagonal pattern of the heatmap evidences the high quality of the obtained feature classification.

7.4.2 Biclustering in agriculture

There are currently no applications in the agricultural field for biclustering techniques. The reason might be the fact that biclustering techniques are used only in recent years, in which they have been mainly applied to gene expression analysis.

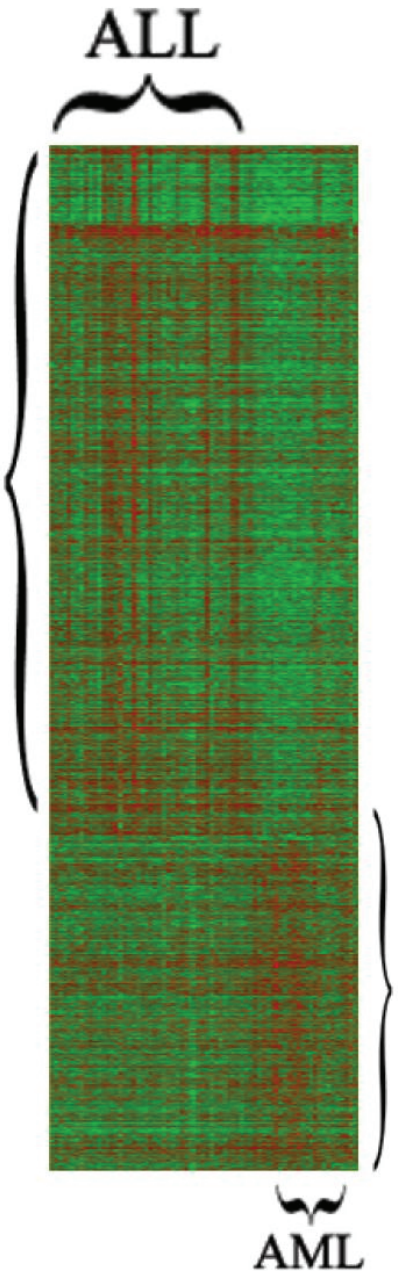

Fig. 7.2 The partition found in biclusters separating the ALL samples and the AML samples.

Tissue type	Abbreviation	Number of samples
Blood	BD	1
Brain	BRA	11
Breast	BRE	2
Colon	CO	1
Cervix	CX	1
Endometrium	ENDO	2
Esophagus	ES	1
Kidney	KI	6
Liver	LI	6
Lung	LU	6
Muscle	MU	6
Myometrium	MYO	2
Ovary	OV	2
Placenta	PL	2
Prostate	PR	4
Spleen	SP	1
Stomach	ST	1
Testes	TE	1
Vulva	VU	3

Fig. 7.3 Tissues from the HuGE Index set of data.

In fact, biclustering was introduced in the literature in 1972 by Hartigan [103], but only later, in 2000, Cheng and Church took the idea and applied it to expression data [47]. Another reason for the non-use of biclustering in agriculture may be the complexity of the method. As usual, scientists who are expert in fields different from numerical analysis and computer science tend to use easier solutions. This is one of the reasons why methods such as k-means are applied more than neural networks or support vector machines in applied fields.

However, it is our opinion that biclustering may provide good results if applied to agricultural problems. Let us take as example the problem considered in Section 3.5.1, where wine fermentation problems are predicted by a k-means approach. In this example, each sample is represented as a vector having as components some compounds measured in the wine during the fermentation process. The goal is to predict wine fermentation problems that may occur using information about the compounds measured not later than 3 days after the start of the fermentation process. The clustering algorithm used provides a partition of the samples but no considerations are made about the compounds that are responsible for these partitions. Biclustering might also provide this kind of information. If the feature is known, a particular compound in this case that is associated to a cluster of samples, then such samples are similar because of that feature. In this application, besides discovering patterns that signal fermentation problems, the compounds that are more responsible for such problems can be located. This may help the work of the enologist when his intervention is required to correct the fermentation process.

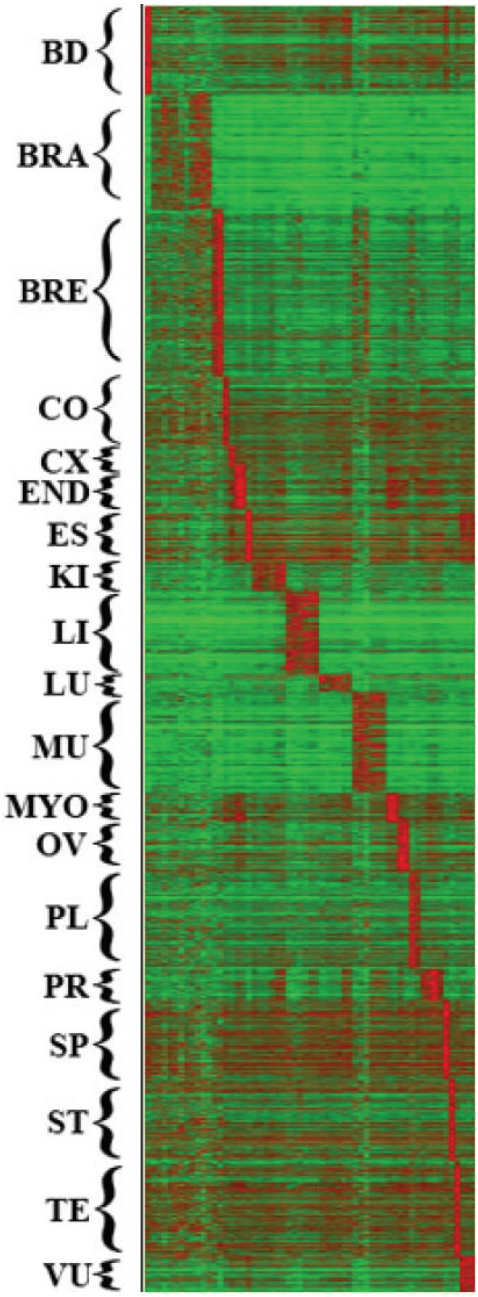

Fig. 7.4 The partition found in biclusters of the tissues in the HuGE Index set of data.

Biclustering can be applied even to other applications discussed in the other chapters of the book. In particular, when a training set is available, and classification techniques can be used, then a partition in biclusters of the data can be found before the classification technique is applied. This can be done using the rule (7.2). When the biclusters are found, each class in the original training set is associated to a cluster of features. This allows one to find out which are the features responsible for grouping a subset of samples in a certain class. In order to be sure that each feature is actually assigned to the right class, the partition in biclusters has to be consistent. The consistency can be checked by applying the rule (7.3) and checking if the original classification in the training set is found again. In the case the partition is not consistent, then some of the features need to be discarded. This task could be done by hand if the classification problem is not so large. Otherwise, the optimization problem (7.11)–(7.10) needs to be solved.

Note that, once the samples in a testing set have been classified by using a classification technique, the rule (7.3) can be applied to it and another partition in biclusters can be found. The classification technique tries to reproduce the classification in the training set on unclassified samples. Therefore, choosing a certain class, the corresponding bicluster in the training set and the one in the testing set should be similar. This may also be used for validating the data mining technique used.

7.5 Exercises

In this section some exercises related to biclustering are presented.

1. Consider the matrix
$$A = \begin{pmatrix} 1 & 2 & 3 & -4 & 5 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 2 & 0 \\ -1 & 3 & 1 & 0 & 2 \\ 3 & -1 & 1 & 2 & 1 \end{pmatrix}.$$
Locate a bicluster with constant row values having dimension 2×2.
2. Consider 6 samples in a three-dimensional space:
$$x_1 = (7, 0, 0), \quad x_2 = (5, 0, 0), \quad x_3 = (0, 1, 0),$$
$$x_4 = (0, 3, 0), \quad x_5 = (0, 0, 1), \quad x_6 = (0, 0, 5).$$
Suppose that they are assigned to 3 clusters as follows:
$$x_1 \in S_1, \quad x_2 \in S_1, \quad x_3 \in S_2, \quad x_4 \in S_2, \quad x_5 \in S_3, \quad x_6 \in S_3.$$
By using the rule (7.2), find a partition of the features used for representing the three-dimensional points. Then, define a partition of the points in biclusters.
3. Verify that the partition in biclusters obtained in the previous exercise is consistent.

4. Consider 4 samples in a three-dimensional space:

$$x_1 = (1, 2, 3), \quad x_2 = (2, 3, 4), \quad x_3 = (3, 4, 2), \quad x_4 = (4, 5, 1).$$

   Suppose that

$$x_1 \in S_1, \quad x_2 \in S_1, \quad x_3 \in S_2, \quad x_4 \in S_2.$$

   Find a partition in biclusters by using the rule (7.2) and check if the biclustering is consistent.
5. Provide an example of partition in biclusters of a given set of data which is α-consistent but not consistent for a certain α value.