Chapter 2

Model-Based Co-Clustering

The simplest co-clustering approach is to simultaneously perform clustering of rows and columns using a partition z of the set I of rows and a partition w of the set J of columns. In the terminology of Tucker [TUC 64], this approach can be characterized as seeking a partition on both modes of a matrix. As for the partitioning clustering situation, here the most frequent approach is the metric approach that consists of defining a clustering criterion and then finding an algorithm optimizing this criterion. In the first section, we develop a general framework for an approach of this kind proposed by Govaert [GOV 83, GOV 95].

2.1. Metric approach

In the context of exploratory analysis, the aim of co-clustering algorithms is to provide an easily interpretable summary, to be able to jointly use factorial methods and clustering methods, to have similar methods adapted to the main data type, and to obtain classical clustering methods when trying to classify one of the two sets. To achieve this

Co-Clustering: Models, Algorithms and Applications Gérard Govaert and Mohamed Nadif
© 2014 ISTE Ltd. Published 2014 by ISTE Ltd.

aim, the main idea of the proposed methodology is to summarize the initial matrix x by a smaller matrix a defined from a couple of partitions z and w of I and J, having the same structure as the initial matrix. For instance, a 1,000 × 200 binary matrix will be summarized by a 10 × 5 binary matrix or a 1,000 × 200 contingency table by a 10 × 5 contingency table. More precisely, the summarization will be made as shown in Figure 2.1 and the function s will depend on the type of data: for binary data, it will be the majority value; for a contingency table, it will be the sum; and for continuous data, it will be the mean.

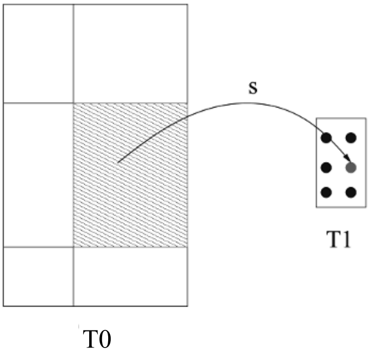

Figure 2.1. *Aggregation of data matrix in summary matrix using co-clustering*

In addition to facilitating the interpretation of results, this approach allows us to define the objective function more easily: the objective function $\mathrm{W}(\mathrm{z}, \mathrm{w}, \mathrm{a})$ will be defined using a function Δ that measures the difference between the two matrices x and a. A more precise definition will depend on the nature of the data.

Several algorithms are able to find a local optimum of the objective function. For instance, Govaert [GOV 83, GOV 95] proposed an iterative algorithm that defines a sequence of partition pairs (z, w). Starting from an initial partition pair,

(z, w), the following procedure is applied: one of these partitions is fixed and a better partition of the other set is searched for. Then, the resulting partition is fixed and a better partition of the first set is searched for. These two steps are repeated until convergence. For finding the better partition at each step, dynamic cluster method [DID 79, CEL 89] can be used. Since this last method is also iterative, the co-clustering algorithms have two levels of iterations. The properties of these types of algorithms are simplicity, speed of convergence and scalability. The drawbacks are due to the fact that they only provide a local optimum. This approach will be discussed in more detail in the following chapters and will lead, respectively, to algorithms CROBIN, CROKI2 and CROEUC for binary, contingency and continuous data.

2.2. Probabilistic models

Like the classical clustering, the co-clustering methods described in the previous section are heuristic techniques. To avoid this empirical approach, a number of different probabilistic clustering methods have been proposed, but the use of adapted co-clustering models densities described in this section seems attractive for several reasons: it corresponds to the intuitive idea of a population composed of several blocks, it is strongly linked to classical methods, and it is able to handle a wide variety of special situations in a more or less natural way. These kinds of models are based on a conditional independence assumption that we describe hereafter.

CONDITIONAL INDEPENDENCE.– Given an $n \times d$ data matrix x defined on two sets I and J, it is assumed that there exists a partition z on I and a partition w on J such that the univariate random variables x_{ij} are conditionally independent knowing z and w with a parameterized pdf

$f(x_{ij};\alpha_{k\ell})$ if the row i belongs to the cluster k and the column j belong to the cluster ℓ. Thus, the conditional pdf of x, knowing z and w, can be expressed as

$$\prod_{i,j} f(x_{ij};\alpha_{z_i w_j}) = \prod_{i,j,k,\ell} \{f(x_{ij};\alpha_{k\ell})\}^{z_{ik}w_{j\ell}}. \qquad [2.1]$$

From this hypothesis, three situations can be considered depending on whether I and J are assumed to be random samples or not.

Block model: in this situation, the sets I and J are not random samples and statistical units are the elements $\boldsymbol{x}_{ij}$ of the data matrix. Therefore, this model is parameterized by z, w and $\boldsymbol{\alpha}$. These parameters are collectively represented by $\boldsymbol{\theta}$ and the pdf of the data x for this model can be written as

$$f(\mathrm{x};\boldsymbol{\theta}) = \prod_{i,j} f(x_{ij};\alpha_{z_i w_j}) = \prod_{i,j,k,\ell} \{f(x_{ij};\alpha_{k\ell})\}^{z_{ik}w_{j\ell}},$$

where $\boldsymbol{\alpha} = (\alpha_{k\ell}; k = 1, \dots g, \ell = 1, \dots, m)$. This model will be studied in Chapter 4 which deals with contingency tables.

Mixture model: in this situation, only the set J of rows is regarded as a random sample and the statistical units are rows $\boldsymbol{x}_i$ of the data matrix. Moreover, these statistical units and the row labels, which are latent variables, are assumed to be independent. Therefore, this model is parameterized by $\boldsymbol{\pi} = (\pi_1, \dots, \pi_g)$ where $\pi_k = P(z_{ik} = 1)$, z, w and $\boldsymbol{\alpha}$ are collectively represented by $\boldsymbol{\theta}$. The pdf of the data x can be written as

$$f(\mathrm{x};\boldsymbol{\theta}) = \prod_{i} \left(\sum_{k} \pi_k \prod_{j,\ell} \{f(x_{ij};\alpha_{k\ell})\}^{w_{j\ell}} \right),$$

which is a classical mixture model where, conditional to the component, the variables are independent. This model will be studied in Chapter 5 dealing with continuous variables.

Latent block model: in this last situation, the two sets I and J are considered as random samples and the row and column labels become latent variables. The statistical unit is therefore the data matrix, and the remaining chapter will be devoted to this model.

2.3. Latent block model

2.3.1. *Definition*

To embed the co-clustering in a probabilistic framework, Govaert and Nadif [GOV 03, GOV 05, GOV 06, GOV 07, GOV 08, GOV 10] proposed the latent block model (LBM) and considered different distributions. This model is based on the following assumptions:

– Conditional independence defined in section 2.2.

– Independent latent variables: the labelings $\mathbf{z}_1, \ldots, \mathbf{z}_n, \mathbf{w}_1, \ldots, \mathbf{w}_d$ are considered as latent variables and assumed to be independent

$$p(\mathbf{z}, \mathbf{w}) = p(\mathbf{z})p(\mathbf{w}),\ p(\mathbf{z}) = \prod_i p(z_i) \text{ and } p(\mathbf{w}) = \prod_j p(w_j). \quad [2.2]$$

– For all i, the distribution of $p(z_i)$ is the categorical distribution $\mathcal{M}(\pi_1, \ldots, \pi_g)$ and does not depend on i. Similarly, for all j, the distribution of $p(w_j)$ is the categorical distribution $\mathcal{M}(\rho_1, \ldots, \rho_m)$ and does not depend on j.

Therefore, the parameter of the LBM is

$$\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\alpha}),$$

with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_m)$ where $(\pi_k = P(z_{ik} = 1), k = 1, \ldots, g)$, $(\rho_\ell = P(w_{j\ell} = 1), \ell = 1, \ldots, m)$ are the mixing proportions and $\boldsymbol{\alpha} = (\alpha_{k\ell}; k = 1, \ldots g, \ell = 1, \ldots, m)$ where $\alpha_{k\ell}$ is the parameter of the distribution of block k, ℓ.

Denoting by $\mathcal{Z}$ and $\mathcal{W}$ the sets of possible labels $\mathbf{z}$ for I and w for J, the pdf of x can be written as

$$\begin{aligned} f(\mathrm{x};\boldsymbol{\theta}) &= \sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} p(\mathbf{z},\mathrm{w})f(\mathrm{x}|\mathbf{z},\mathrm{w};\boldsymbol{\theta}) && [2.3] \\ &= \sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} \prod_i \pi_{z_i} \prod_j \rho_{w_j} \prod_{i,j} f(x_{ij};\alpha_{z_i w_j}) && [2.4] \\ &= \sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j\ell} \rho_\ell^{w_{j\ell}} \prod_{i,j,k,\ell} \{f(x_{ij};\alpha_{k\ell})\}^{z_{ik}w_{j\ell}}. && [2.5] \end{aligned}$$

The randomized data generation process corresponding to the LBM with parameter $\boldsymbol{\theta} = (\boldsymbol{\pi},\boldsymbol{\rho},\boldsymbol{\alpha})$ can be described as follows:

– Generate the labelings $\mathbf{z} = (z_1,\ldots,z_n)$ into g clusters according to the categorical distribution $\boldsymbol{\pi} = (\pi_1,\ldots,\pi_g)$.

– Generate the labelings $\mathrm{w} = (w_1,\ldots,w_d)$ into m clusters according to the categorical distribution $\boldsymbol{\rho} = (\rho_1,\ldots,\rho_m)$.

– Generate for $i = 1,...,n$ and $j = 1,...,d$ a real value x_{ij} according to the distribution $f(.;\boldsymbol{\alpha}_{z_i w_j})$.

This model can be represented by a graphical model shown in Figure 2.2.

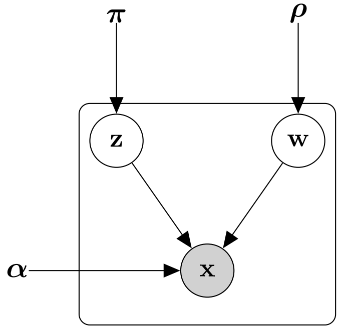

Figure 2.2. *Latent block model as a graphical model*

According to the type of data (binary, contingency and continuous) various monodimensional distribution functions can be used. In the following chapters, three distributions will be studied: Bernoulli distributions for binary data, Poisson distributions for contingency data and Gaussian distributions for continuous data, but the model can be extended by taking into account, for example, the outliers by using Student's distributions.

Note that the LBM is dramatically more parsimonious than using a classical mixture model on each of the two sets of rows and columns: for example, with $n = 1{,}000$ rows and $d = 500$ columns and equal class probabilities $\pi_k = 1/g$ and $\rho_\ell = 1/m$, if we need to cluster the data matrix into $g = 4$ clusters of rows and $m = 3$ clusters of columns, the Poisson LBM will involve the estimation of 12 parameters $(\alpha_{k\ell}, k = 1, \ldots, 4, \ell = 1, \ldots, 3)$, instead of $(4 \times 500 + 3 \times 1{,}000) = 5{,}000$ parameters with two mixture models applied on the two sets separately.

Links with other models such as the mixture model, the latent Dirichlet allocation (LDA) model [BLE 03] or the stochastic block model can be established. We study the link with the classical mixture model in the next section.

2.3.2. *Link with the mixture model*

It is natural to ask about the relationship between the LBM and the classical mixture model. To this end, it suffices to express the pdf of the data x conditionally on the partition w. This takes the form

$$\begin{aligned} f(\mathbf{x}|\mathbf{w}) &= \sum_{\mathbf{z}} p(\mathbf{z}) f(\mathbf{x}|\mathbf{z}, \mathbf{w}) = \sum_{\mathbf{z}} p(\mathbf{z}) \prod_{i,j} f(x_{ij}, \alpha_{z_i w_j}) \\ &= \sum_{\mathbf{z}} \prod_i p(z_i) \prod_{i,j} f(x_{ij}, \alpha_{z_i w_j}) = \sum_{\mathbf{z}} \prod_i \left(p(z_i) \prod_j f(x_{ij}, \alpha_{z_i w_j}) \right) \\ &= \prod_i \sum_k \left(\pi_k \prod_j f(x_{ij}, \alpha_{k w_j}) \right), \end{aligned}$$

which leads to the following properties:

– the component x_{ij} of row i are conditionally independent on z_{ik};

– the rows x_i of the data matrix x are conditionally independent on w;

– conditional on the partition w, the pdf of the data x is a mixture model;

and, symmetrically:

– the components x_{ij} of column i are conditionally independent on $w_{j\ell}$;

– the columns x_j of the data matrix x are conditionally independent on z;

– conditional on the partition z, the pdf of the data x is a mixture model.

2.3.3. *Log-likelihoods*

In the following, two log-likelihoods will be used: the classical log-likelihood $\mathrm{L}(\boldsymbol{\theta})$, which is the logarithm of the sample distribution $f(\mathrm{x};\boldsymbol{\theta})$ seen as a function of $\boldsymbol{\theta}$, and the *complete-data log-likelihood* function, which is the logarithm of the sample distribution of the latent data, or complete data. For the LBM, the complete data are taken to be the vector $(\mathrm{x}, \mathrm{z}, \mathrm{w})$ where unobservable vectors z and w are the labels. The complete data log-likelihood can therefore be written as

$$\begin{aligned}
\mathrm{L_C}(\mathbf{z},\mathbf{w},\boldsymbol{\theta}) &= \mathrm{L}(\boldsymbol{\theta};\mathbf{x},\mathbf{z},\mathbf{w}) = \log f(\mathbf{x},\mathbf{z},\mathbf{w};\boldsymbol{\theta})\\
&= \log\{p(\mathbf{z};\boldsymbol{\theta})p(\mathbf{w};\boldsymbol{\theta})f(\mathbf{x}|\mathbf{z},\mathbf{w};\boldsymbol{\alpha})\}\\
&= \log p(\mathbf{z};\boldsymbol{\theta}) + \log p(\mathbf{w};\boldsymbol{\theta}) + \log f(\mathbf{x}|\mathbf{z},\mathbf{w};\boldsymbol{\alpha})\\
&= \log \prod_{i,k} \pi_k^{z_{ik}} + \log \prod_{j,\ell} \rho_\ell^{w_{j\ell}} + \log f(\mathbf{x}|\mathbf{z},\mathbf{w};\boldsymbol{\alpha}),
\end{aligned}$$

and, finally, the complete-data log-likelihood becomes

$$\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = \sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell + \sum_{i,j,k,\ell} z_{ik} w_{j\ell} \log f(x_{ij}; \alpha_{k\ell}). \qquad [2.6]$$

Note that this complete-data log-likelihood divides into three terms: the first term depends on proportions of row clusters, the second term depends on proportions of column clusters and the third term depends on the pdf of each block or bicluster.

2.3.4. *A complex model*

The LBM involves a double missing data structure, namely z and w, which makes statistical inference more difficult than for the standard mixture model. In this section, we list some of these difficulties.

2.3.4.1. *Computing the likelihood*

Even for small tables, computing this likelihood (or its logarithm) is difficult. For instance, with a data matrix 20×20 with $g = 2$ and $m = 2$, it requires the calculation of $gn \times md = 1{,}600$ terms. As a consequence, even though the parameter $\boldsymbol{\theta}$ is properly estimated, computing penalized model selection criteria such as the Akaike information criterion (AIC) or Bayesian information criterion (BIC) is challenging. Moreover, these difficulties of computing relevant model selection criteria are increased by the fact that the LBM statistical units could be defined in several different ways.

2.3.4.2. *Classification step*

With the parameter $\boldsymbol{\theta}$ being fixed, the problem is to determine the "best" partitions z and w, i.e. the pair of partitions z, w maximizing the posterior probability $f(\mathrm{z}, \mathrm{w}|\mathrm{x}, \boldsymbol{\theta})$ (MAP). Knowing that

$$f(\mathrm{z}, \mathrm{w}|\mathrm{x}, \boldsymbol{\theta}) = \frac{f(\mathrm{x}, \mathrm{z}, \mathrm{w}; \boldsymbol{\theta})}{f(\mathrm{x}; \boldsymbol{\theta})},$$

the couple (z, w) is obtained by maximizing the complete-data log-likelihood $\mathrm{L_C}(\boldsymbol{\theta}, \mathrm{z}, \mathrm{w}) = \log f(\mathrm{x}, \mathrm{z}, \mathrm{w}; \boldsymbol{\theta})$ in z and w. This result generalizes the case of the simple mixing model where the MAP is obtained by maximizing $\mathrm{L_C}(\boldsymbol{\theta}, \mathrm{z})$ in z, i.e. $\log \prod_i f(\boldsymbol{x}_i, z_i)$, which leads to placing each $\boldsymbol{x}_i$ in the class z_i that maximizes

$$f(\boldsymbol{x}_i, z_i) \propto \frac{f(\boldsymbol{x}_i, z_i)}{f(\boldsymbol{x}_i)} = f(z_i|\boldsymbol{x}_i).$$

Unfortunately, in the latent block situation, the maximization of the complete-data log-likelihood $\mathrm{L_C}(\mathrm{z}, \mathrm{w}, \boldsymbol{\theta})$ does not lead to explicit formulas and iterative algorithms are needed.

2.3.4.3. *Error rate*

One characteristic of a mixture model is the degree of mixing among the components. In the classical situation, this concept of cluster separation can be visualized, for instance, by using principal component analysis, but this concept of cluster separation is difficult to apply to the LBMs. Another solution is to compute the true error rate associated with the model, which is defined as the expectation of the misclassification probability $\mathbb{E}(\delta((\mathrm{z}, \mathrm{w}), d(\mathrm{x}))$ where z, w and x are the random variables associated with the LBM, d is the optimal Bayes' rule $d(\mathrm{x}) = (\mathrm{z}', \mathrm{w}') = \arg\max_{\mathbf{z},\mathbf{w}} p(\mathrm{z}, \mathrm{w}|\mathrm{x})$ associated with this model and δ is the error rate.

This expectation is generally difficult to compute theoretically, and Monte Carlo simulations are used to estimate it by the proportion of misclassification, for instance in the classical clustering situation, between the partition simulated with those we obtained by applying a classification step. This proportion of misclassification can be defined as follows: if C is the confusion matrix between the two partitions, relabel the components of the partition $\mathbf{z}'$ such that the trace of matrix C is maximal, then compute $e(\mathbf{z},\mathbf{z}') = 1 - \frac{1}{n}\sum_{i,k} z_{ik}z'_{ik}$. This definition can be extended to the comparison of two pairs of partitions $\mathbf{u} = (\mathbf{z},\mathbf{w})$ and $\mathbf{u}' = (\mathbf{z}',\mathbf{w}')$ as follows

$$\delta(\mathbf{u},\mathbf{u}') = \delta((\mathbf{z},\mathbf{w}),(\mathbf{z}',\mathbf{w}')) = 1 - \frac{1}{nd}\sum_{i,j,k,\ell} u_{ijk\ell}u'_{ijk\ell},$$

where $u_{ijk\ell} = z_{ik}w_{j\ell}$ and $u'_{ijk\ell} = z'_{ik}w'_{j\ell}$, and it can be shown that

$$\delta((\mathbf{z},\mathbf{w}),(\mathbf{z}',\mathbf{w}')) = e(\mathbf{z},\mathbf{z}') + e(\mathbf{w},\mathbf{w}') - e(\mathbf{z},\mathbf{z}') \times e(\mathbf{w},\mathbf{w}').$$

However, this approach is challenging in the LBM situation:

– The classification step is not direct as in a classical mixture model situation, and an iterative algorithm must be used (see section 2.3.4.2).

– The error rate depends not only on the parameter $\boldsymbol{\theta}$ but also on the sizes n and d of the data array, which makes Monte Carlo estimation of the error rate impractical. This unusual phenomenon can be seen in Figure 2.3: for a given distribution, the error rates decrease as the table size increases. The error decrease reflects a property of the table distribution: its size plays an important role in setting the Bayes' risk, i.e. the intrinsic difficulty of the task. To understand this phenomenon, consider the representation of an $n \times d$ table as n d-dimensional vectors. The overall dissimilarity among vectors will grow as d grows. Figure 2.4

shows the principal components of such vectors, extracted from two tables with $d = 50$ in the left plot and $d = 500$ in the right plot. The distribution describing the classes and their probabilities is identical in both plots, but while the clusters in dimension $d = 50$ highly overlap, they are well separated in $d = 500$. Figures 2.3 and 2.4 illustrate a systematic but little-known property of block clustering: the distribution of the table entries being fixed, the Bayes' risk decreases with the table size. More formal arguments, already developed for the more constrained stochastic block model [CEL 12], could be transposed onto co-clustering. Intuitively, this decrease can be understood by considering that the table enlargement in one dimension results in more redundancy in the other dimension.

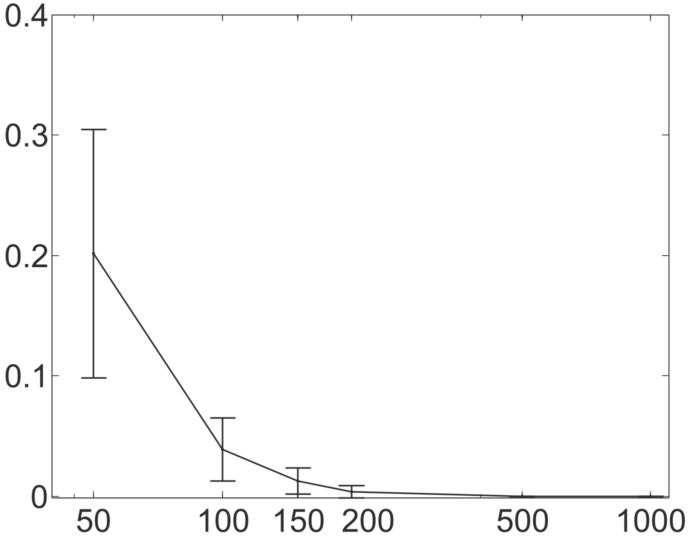

Figure 2.3. *Error rates and interquartile range versus table size for square data tables ($m = n$) with 3×3 clusters whose entries are generated from the same distribution*

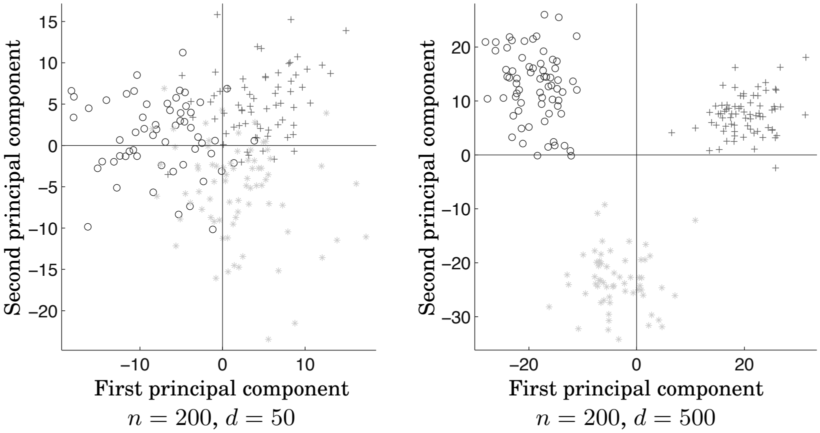

Figure 2.4. *Projections of the rows of two data tables on the first two column eigenvectors. The distribution of table entries is identical, but table sizes differ, with* $n = 200$ *rows and* $d = 50$ *(left) or* $d = 500$ *(right)*

2.4. Maximum likelihood estimation and algorithms

As for the classical mixture model, the maximization of the likelihood is not straightforward and the expectation–maximization (EM) algorithm [DEM 77], which maximizes the log-likelihood $\mathrm{L}(\boldsymbol{\theta})$ iteratively by maximizing the conditional expectation of the *complete-data log-likelihood* given a previous current estimate $\boldsymbol{\theta}^{(c)}$ and the data x, can be considered. Unfortunately, difficulties arise owing to the dependence structure in the model. The LBM involves a double missing data structure, namely z and w, which makes statistical inference more difficult than for the standard mixture model. As previously seen, computing the likelihood [2.3] or its logarithm is difficult. In the same manner, deriving the maximum likelihood estimator with the EM algorithm is challenging. As a matter of fact, the E step requires the computation of the joint conditional distributions of the missing labels $p(z_{ik}w_{j\ell} = 1|\mathrm{x}; \boldsymbol{\theta})$ for $i \in I$, $j \in J$, $k \in K$ and $\ell \in L$, $\boldsymbol{\theta}$ being a current value of the parameter. Thus, the E step, which computes the conditional expectation of the complete data log-likelihood given a previous current estimate $\boldsymbol{\theta}^{(c)}$ and x

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)}) = \sum_{i,k} p(z_{ik} = 1|\mathrm{x}, \boldsymbol{\theta}^{(c)}) \log \pi_k + \sum_{j,\ell} p(w_{j\ell} = 1|\mathrm{x}, \boldsymbol{\theta}^{(c)}) \log \rho_\ell$$

$$+ \sum_{i,j,k,\ell} p(z_{ik} w_{j\ell} = 1|\mathrm{x}, \boldsymbol{\theta}^{(c)}) \log f(x_{ij}; \alpha_{k\ell}),$$

involves computing too many terms that cannot be factorized, such as for a standard mixture, due to the conditional dependence on the observations of the row and column labels.

To solve this problem, an approximation using the Neal and Hinton interpretation [NEA 98] of the EM algorithm can be proposed. In Chapter 1, it was seen that, for mixture models, the EM algorithm can be viewed as an alternating algorithm. To extend this property to all models that can be dealt with by using the EM algorithm, Neal and Hinton propose to define the following function

$$\mathrm{F_C}(R; \boldsymbol{\theta}) = \mathbb{E}_R(\mathrm{L_C}(\mathbf{y}; \boldsymbol{\theta})|\mathrm{x}) + \mathrm{H}(R), \qquad [2.7]$$

where y is the complete data, R is a probability over the space of complete data and $\mathrm{H}(R)$ is the entropy of the distribution R. We can also relate $\mathrm{F_C}$ to the Kullback–Liebler divergence between R and $P_{\boldsymbol{\theta}}$ defined by $P_{\boldsymbol{\theta}}(\mathrm{y}) = p(\mathrm{y}|\mathrm{x}, \boldsymbol{\theta})$ as follows

$$\mathrm{F_C}(R, \boldsymbol{\theta}) = \mathrm{L}(\theta) - \mathrm{KL}(R, P_{\boldsymbol{\theta}}). \qquad [2.8]$$

The alternated optimization of the $\mathrm{F_C}$ function is therefore simple to set up: for fixed $\boldsymbol{\theta}$, the maximization of equation [2.8] yields to the minimization of $\mathrm{KL}(R, \mathbb{P}_\theta)$ and therefore to $R = P_{\boldsymbol{\theta}}$; for fixed R, the maximization of equation [2.7] shows that $\boldsymbol{\theta}$ must maximize the expectation $E_R(\mathrm{L_C}(\mathrm{y}, \boldsymbol{\theta})|\mathrm{x})$. These two steps are precisely those of the EM algorithm. Moreover, after the first step, we have $\mathrm{F_C}(R, \boldsymbol{\theta}) = \mathrm{F_C}(P_{\boldsymbol{\theta}}, \boldsymbol{\theta}) = L(\boldsymbol{\theta})$, which shows that each iteration increases the log-likelihood. The relation between the log-likelihood and the fuzzy criterion $\mathrm{F_C}$ can be written as $\mathrm{L}(\boldsymbol{\theta}) = \arg\max_R \mathrm{F_C}(R, \boldsymbol{\theta})$.

For the LB model, R is a distribution on $\mathcal{Z} \times \mathcal{W}$ and the first step leads to taking $R = P(\mathbf{z}, \mathrm{w}|\mathrm{x}, \boldsymbol{\theta})$, which is, obviously, always intractable but, to derive the maximum likelihood estimate of $\boldsymbol{\theta}$, two approximations can be proposed: a variational EM approach and a classification EM approach.

2.4.1. *Variational EM approach*

A variational approximation of the EM algorithm, denoted as VEM in this book, can be proposed by imposing that the distribution R of the labels is assumed to be independent

$$R(\mathbf{z}, \mathrm{w}) = R(\mathbf{z})R(\mathrm{w}), \quad R(\mathbf{z}) = \prod_i q(z_i)$$

and $$R(\mathrm{w}) = \prod_j q(w_j). \qquad [2.9]$$

Denoting $\widetilde{z}_{ik} = R(z_{ik} = 1)$, $\widetilde{\mathrm{w}}_{j\ell} = R(w_{j\ell} = 1)$, $\widetilde{z}_{.k} = \sum_i \widetilde{z}_{ik}$ and $\widetilde{\mathrm{w}}_{.\ell} = \sum_j \widetilde{\mathrm{w}}_{j\ell}$, the expectation of complete-data log-likelihood and the entropy become

$$\mathbb{E}_R(\mathrm{L_C}(\mathbf{y};\boldsymbol{\theta})|\mathbf{x}) = \sum_k \widetilde{z}_{.k} \log \pi_k + \sum_\ell \widetilde{\mathrm{w}}_{.\ell} \log \rho_\ell + \sum_{i,j,k,\ell} \widetilde{z}_{ik}\widetilde{\mathrm{w}}_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$$

and

$$\mathrm{H}(P) = \mathrm{H}(\widetilde{\mathbf{z}}) + \mathrm{H}(\widetilde{\mathbf{w}}),$$

where $\mathrm{H}(\widetilde{\mathbf{z}}) = -\sum_{i,k} \widetilde{z}_{ik} \log \widetilde{z}_{ik}$ and $H(\widetilde{\mathbf{w}}) = -\sum_{j,\ell} \widetilde{\mathrm{w}}_{j\ell} \log \widetilde{\mathrm{w}}_{j\ell}$. Therefore, the fuzzy clustering criterion [2.7] can be written as

$$\mathrm{F_C}(\widetilde{\mathbf{z}}, \widetilde{\mathbf{w}}; \boldsymbol{\theta}) = \mathrm{L_C}(\widetilde{\mathbf{z}}, \widetilde{\mathbf{w}}, \boldsymbol{\theta}) + \mathrm{H}(\widetilde{\mathbf{z}}) + \mathrm{H}(\widetilde{\mathbf{w}}), \qquad [2.10]$$

where $\mathrm{L_C}$ is the fuzzy complete-data log-likelihood associated with the block latent model depending on three terms as in equation [2.6] but weighted by posterior probabilities

$$\begin{aligned}\mathrm{L_C}(\widetilde{\mathbf{z}}, \widetilde{\mathbf{w}}; \boldsymbol{\theta}) = {} & \sum_k \widetilde{z}_{.k} \log \pi_k + \sum_\ell \widetilde{\mathrm{w}}_{.\ell} \log \rho_\ell \\ & + \sum_{i,j,k,\ell} \widetilde{z}_{ik} \widetilde{\mathrm{w}}_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell}).\end{aligned}$$

The objective of the variational approach is therefore to maximize the function $\mathrm{F_C}$. In doing so, we have replaced the maximization of the likelihood by the maximization of an approximation of this likelihood defined by

$$\widetilde{\mathrm{L}}(\boldsymbol{\theta}) = \arg\max_{\widetilde{\mathbf{z}}, \widetilde{\mathbf{w}}} \mathrm{F_C}(\widetilde{\mathbf{z}}, \widetilde{\mathbf{w}}, \boldsymbol{\theta}).$$

The maximization of the function $\mathrm{F_C}(\widetilde{\mathbf{z}}, \widetilde{\mathbf{w}}, \boldsymbol{\delta})$ can be obtained by alternating the following three computations:

1) $\widetilde{\mathbf{z}} = \arg\max_{\widetilde{\mathbf{z}}} F_C(\widetilde{\mathbf{z}}, \widetilde{\mathbf{w}}, \boldsymbol{\theta})$: equations [2.10] and [2.11] lead, for all i, to

$$\widetilde{\mathbf{z}} = \arg\max_{\widetilde{\mathbf{z}}} \sum_k \widetilde{z}_{ik} \left(\log \pi_k + \sum_{j,\ell} \widetilde{\mathrm{w}}_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell}) \right)$$

under the constraint $\sum_k \widetilde{z}_{ik} = 1$ and therefore to

$$\widetilde{z}_{ik} \propto \pi_k \exp(\sum_{j,\ell} \widetilde{\mathrm{w}}_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})).$$

2) $\widetilde{\mathbf{w}} = \arg\max_{\widetilde{\mathbf{z}}} F_C(\widetilde{\mathbf{z}}, \widetilde{\mathbf{w}}, \boldsymbol{\theta})$: similarly, we obtain

$$\widetilde{\mathrm{w}}_{j\ell} \propto \rho_\ell \exp(\sum_{i,j} \widetilde{z}_{ik} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})).$$

3) $\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} F_C(\widetilde{\mathbf{z}}, \widetilde{\mathbf{w}}, \boldsymbol{\theta})$: equations [2.10] and [2.11] lead to

$$\boldsymbol{\pi} = \arg\max_{\boldsymbol{\pi}} \sum_k \widetilde{z}_k \log \pi_k \quad \text{and} \quad \boldsymbol{\rho} = \arg\max_{\boldsymbol{\rho}} \sum_\ell \widetilde{\mathrm{w}}_\ell \log \rho_\ell$$

and therefore to

$$\pi_k = \frac{\widetilde{z}_{.k}}{n} \quad \forall k \qquad \text{and} \qquad \rho_\ell = \frac{\widetilde{\mathrm{w}}_{.\ell}}{d} \quad \forall \ell.$$

For the block parameters, the computations of the $\alpha_{k\ell}$'s, defined by $\arg\max_{\alpha_{k\ell}} \sum_{i,j} \widetilde{z}_{ik}\widetilde{\mathrm{w}}_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$ $\forall k, \ell$, will depend on the distribution of the component and will be studied in the following chapters.

Different strategies of alternated optimization can be proposed. One of them, denoted as LBVEM algorithm and described in algorithm [2.1], has been proved to give relevant estimates of the LBM model in different contexts; see, for instance, [GOV 08] for continuous or binary data.

After we fit the LBM to estimate $\boldsymbol{\theta}$, we can give an outright or hard clustering of these data by assigning each observation to the component of the mixture which it has the highest posterior probability of belonging to, that is to say to compute the maximum *a posteriori* (MAP). Unfortunately, this maximization does not lead to explicit formulas and iterative algorithms are needed. Here, a simple solution is to assign each row or column to the component which maximizes the probabilities $\widetilde{z}_{ik}$ or $\widetilde{\mathrm{w}}_{j\ell}$ obtained at the end of the VEM algorithm.

Algorithm 2.1 LBVEM

```
input: $\mathrm{x}$, $g$, $m$
initialization: $\widetilde{\mathrm{z}}$, $\widetilde{\mathrm{w}}$, $\pi_k = \frac{\widetilde{z}_{.k}}{n}$, $\rho_\ell = \frac{\widetilde{\mathrm{w}}_{.\ell}}{d}$,
$\alpha_{k\ell} = \arg\max_{\alpha_{k\ell}} \sum_{i,j} \widetilde{z}_{ik}\widetilde{\mathrm{w}}_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$
repeat
  repeat
    step 1. $\widetilde{z}_{ik} \propto \pi_k \exp(\sum_{j,\ell} \widetilde{\mathrm{w}}_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell}))$
    step 2. $\pi_k = \frac{\widetilde{z}_{.k}}{n}$,
    $\alpha_{k\ell} = \arg\max_{\alpha_{k\ell}} \sum_{i,j} \widetilde{z}_{ik}\widetilde{\mathrm{w}}_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$
  until convergence
  repeat
    step 3. $\widetilde{\mathrm{w}}_{j\ell} \propto \rho_\ell \exp(\sum_{i,k} \widetilde{z}_{ik} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell}))$
    step 4. $\rho_\ell = \frac{\widetilde{\mathrm{w}}_{.\ell}}{d}$,
    $\alpha_{k\ell} = \arg\max_{\alpha_{k\ell}} \sum_{i,j} \widetilde{z}_{ik}\widetilde{\mathrm{w}}_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$
  until convergence
until convergence
return $\boldsymbol{\pi}$, $\boldsymbol{\rho}$, $\boldsymbol{\alpha}$
```

2.4.2. *Classification EM approach*

Another approximation of the EM algorithm can be obtained by replacing, in equation [2.7], the constraint [2.9] with the constraint

$$p(z_{ik} = 1, w_{j\ell} = 1) \in \{0, 1\}.$$

In doing so, the fuzzy partitions $\widetilde{\mathrm{z}}$ and $\widetilde{\mathrm{w}}$ are replaced by the “hard” partitions z and w and, as the entropies $\mathrm{H}(\mathbf{z})$ and $\mathrm{H}(\mathbf{w})$ become zero, the objective of this approach is therefore to maximize the complete-data log-likelihood, also called classification log-likelihood,

$$\mathrm{L_C}(\mathbf{z}, \mathbf{w}; \boldsymbol{\theta}) = \sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell + \sum_{i,j,k,\ell} z_{ik} w_{j\ell} f(x_{ij}, \alpha_{k\ell}). \qquad [2.11]$$

This optimization can be performed by using the classification EM (CEM) algorithm [CEL 92]. It consists of inserting a classification step between E and M steps. Note that equation [2.11] is just a hard version of the fuzzy version expressed in equation [2.11]. The principal steps of the algorithm, that we refer to as LBCEM, are reported in algorithm 2.2.

Algorithm 2.2 LBCEM

```
input: x, $g$, $m$
initialization: z, w, $\pi_k = \frac{z_{.k}}{n}$, $\rho_\ell = \frac{w_{.\ell}}{d}$,
$\alpha_{k\ell} = \arg\max_{\alpha_{k\ell}} \sum_{i,j} z_{ik} w_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$
repeat
  repeat
    step 1. $z_i = \arg\max_k \left(\sum_{j,\ell} w_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell}) + \log \pi_k\right)$
    step 2. $\pi_k = \frac{z_{.k}}{n}$,
    $\alpha_{k\ell} = \arg\max_{\alpha_{k\ell}} \sum_{i,j} z_{ik} w_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$
  until convergence
  repeat
    step 3. $w_j = \arg\max_\ell \left(\sum_{i,k} z_{ik} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell}) + \log \rho_\ell\right)$
    step 4. $\rho_\ell = \frac{\widetilde{\mathrm{w}}_{.\ell}}{d}$,
    $\alpha_{k\ell} = \arg\max_{\alpha_{k\ell}} \sum_{i,j} z_{ik} w_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$
  until convergence
until convergence
return $\boldsymbol{\pi}$, $\boldsymbol{\rho}$, $\boldsymbol{\alpha}$, z, w
```

2.4.3. *Stochastic EM-Gibbs approach*

An alternative will be to replace the EM algorithm with the Gibbs sampling algorithm that, unlike the EM algorithm, may be implemented accurately without problems. Moreover, this approach, which does not stop at the first encountered fixed point of EM [MCL 07], is a possible way to attenuate the dependence of VEM on its initial values. The difficulty is that it is a stochastic algorithm that requires a large number of iterations and does not directly provide a pointwise estimate.

The basic idea of these stochastic EM algorithms is to incorporate a stochastic step between the E and M steps where the missing data are simulated according to their conditional distribution, knowing the observed data and a current estimate of the model parameters. For the LBM, it is not possible to simulate in a single exercise the missing labels z and w, and a Gibbs sampling scheme is required to simulate the couple (z, w). The SEM-Gibbs algorithm (algorithm 2.3) is a simple adaptation to the LBVEM of the standard SEM algorithm of [CEL 85].

Algorithm 2.3 LBSEM

```
input: x, $g$, $m$
initialization: z, w, $\pi_k = \frac{z_{.k}}{n}$, $\rho_\ell = \frac{w_{.\ell}}{d}$,
$\alpha_{k\ell} = \arg\max_{\alpha_{k\ell}} \sum_{i,j} z_{ik} w_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$
repeat
  repeat
    step 1. $\widetilde{z}_{ik} \propto \pi_k \exp(\sum_{j,\ell} w_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell}))$
    step 1'. simulation of $z_i$ according to $\mathcal{M}(\widetilde{z}_{i1}, \ldots, \widetilde{z}_{ig})$
    step 2. $\pi_k = \frac{z_{.k}}{n}$,
    $\alpha_{k\ell} = \arg\max_{\alpha_{k\ell}} \sum_{i,j} z_{ik} w_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$
  until convergence
  repeat
    step 3. $\widetilde{\mathrm{w}}_{j\ell} \propto \rho_\ell \exp(\sum_{i,k} z_{ik} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell}))$
    step 3'. simulation of $w_j$ according to $\mathcal{M}(\widetilde{\mathrm{w}}_{j1}, \ldots, \widetilde{\mathrm{w}}_{jm})$
    step 4. $\rho_\ell = \frac{w_{.\ell}}{d}$,
    $\alpha_{k\ell} = \arg\max_{\alpha_{k\ell}} \sum_{i,j} z_{ik} w_{j\ell} \log f(x_{ij}, \boldsymbol{\alpha}_{k\ell})$
  until convergence
until convergence
return $\boldsymbol{\pi}$, $\boldsymbol{\rho}$, $\boldsymbol{\alpha}$
```

Note that the formulas of VEM and SEM-Gibbs are essentially the same, except that the probabilities $\widetilde{z}_{ik}$ and $\widetilde{\mathrm{w}}_{j\ell}$

are replaced with binary indicator values z_{ik} and $w_{j\ell}$. However, it is not the only difference between VEM and SEM-Gibbs: while VEM is based on the variational approximation of the LBM, SEM-Gibbs uses no approximation, but runs a Gibbs sampler to simulate the unknown labels with their conditional distribution knowing the observations and a current estimation of the parameters. SEM-Gibbs does not increase the log-likelihood at each iteration. It generates an irreducible Markov chain with a unique stationary distribution that is expected to be concentrated about the maximum likelihood parameter estimate. Thus, a natural estimate of $\boldsymbol{\theta}$ derived from SEM-Gibbs is the mean $\bar{\boldsymbol{\theta}}$ of $(\boldsymbol{\theta}^{(c)}; c = B+1, \ldots, B+C)$ obtained after a burn-in period of length B. Numerical experiments presented in [KER 13] for binary data show that SEM-Gibbs is by far less sensitive to starting values than VEM. Those results have led them to advocate initializing the VEM algorithm with the SEM-Gibbs mean parameter estimate $\bar{\boldsymbol{\theta}}$ to obtain a good approximation of the maximum likelihood estimate for the LBM.

2.5. Bayesian approach

Bayesian inference in statistics can be regarded as a well-grounded tool for regularizing the maximum likelihood estimates in a poorly posed setting. In the LBM setting, Bayesian inference could be considered as useful in avoiding spurious solutions and thus attenuate the "empty cluster" problem. Using Bayesian inference from a regularization perspective, the model parameter may be estimated by maximizing the posterior probability $p(\boldsymbol{\theta}|\mathrm{x})$; it leads to the so-called MAP estimate

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{x}).$$

The Bayes' formula

$$\log p(\boldsymbol{\theta}|\mathbf{x}) = \log p(\mathbf{x}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log p(\mathbf{x})$$

allows us to straightforwardly define an EM algorithm for the computation of the MAP estimate

– The E step relies on the computation of the conditional expectation of the complete log-likelihood $R(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$ as for the maximum likelihood estimator

$$R(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)}) = \mathbb{E}\big(\log p(\mathbf{x}, \mathbf{z}, \mathrm{w}|\boldsymbol{\theta})|\mathbf{x}, \boldsymbol{\theta}^{(c)}\big).$$

– The M step differs in that the objective function for the maximization process is equal to the $R(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$ function augmented by the logarithm of the prior probability

$$\boldsymbol{\theta}^{(c+1)} = \arg\max_{\boldsymbol{\theta}} \big(R(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)}) + \log p(\boldsymbol{\theta})\big).$$

This M Step forces an increase in the log-posterior function $p(\boldsymbol{\theta}|\mathrm{x})$ [MCL 07, Chapter 6, p. 231]. For LBM, the Bayesian approach combined with the variational approximation leads to the V-Bayes algorithm. This approach will be discussed, in more detail for binary and categorical data in Chapter 3.

2.6. Conclusion and miscellaneous developments

In this chapter, we developed the LBM and three algorithms enabling us to estimate the parameters of this model and leading to co-clustering. Note that LBVEM and LBCEM are two iterative algorithms whose convergence in terms of $\mathrm{F_C}(\widetilde{\mathbf{z}}, \widetilde{\mathrm{w}}; \boldsymbol{\theta})$ or $\mathrm{L_C}(\mathbf{z}, \mathrm{w}; \boldsymbol{\theta})$ is guaranteed. They are simple to implement and scalable. Parsimony of the model and its flexibility in adapting to different types of data, as we will see in the remainder of book, are particularly beneficial in the context of data mining. These algorithms also easily avoid the sparsity problem because they work on intermediate matrices that are compressed. We will discuss this point in detail in Chapter 3. The data are called sparse when the proportion of zeros of the matrix is very important. For storing and manipulating such matrices, it is beneficial and often necessary to use specialized algorithms and data structures that take advantage of the sparse structure of the matrix. The sparse data structure represents a matrix in space proportional to the number of non-zero entries, and most of the operations compute sparse results in time proportional to the number of arithmetic operations on non-zeros (see, for instance, [GIL 92]). In this situation, the algorithms described in this chapter are able to take into account this type of data in a simple and effective way.

However, there are still some challenges to be overcome with this model, such as the choice of the number of clusters (g,m). Choosing relevant numbers of clusters in an LBM is obviously very important. However, this model selection problem is more difficult than in a classical mixture model for several reasons. First, there is a couple (g, m) of number of clusters to be selected. Second, penalized likelihood criteria such as AIC or BIC are not directly available since computing the maximized likelihood is not feasible. Third, determining the number of statistical units of an LBM could be questionable. In the next chapter, it will be shown how this problem can be solved.

Before considering the estimation problem, it is important to analyze the model generic identifiability [FRÜ 06, pp. 21–23]. Obviously, LBM, as a mixture model, is not identifiable due to invariance to relabeling the blocks, but it is of no importance when used for the maximum likelihood estimation. The identifiability will depend on the type of data and will be studied in more detail for the binary and categorical situations in the next chapter.