Dipartimento di Statistica, Probabilità e Statistiche Applicate

Università di Roma "La Sapienza

Dottorato di Ricerca in Statistica Metodologica

Tesi di Dottorato XIX Ciclo - 2003/2006

Francesca Martella

MODEL-BASED DOUBLE CLUSTERING FOR HIGH DIMENSIONAL DATA

Supervisor: Prof. Maurizio Vichi

Roma - Dicembre 2006

Università degli Studi di Roma

La Sapienza

Dipartimento di Statistica, Probabilità e Statistiche Applicate

Dottorato di Ricerca in Statistica Metodologica ciclo XIX

MODEL-BASED DOUBLE CLUSTERING FOR HIGH DIMENSIONAL DATA

A thesis in

Statistics

by

Francesca Martella

<francesca.martella@uniroma1.it>

Rome, December, 2006

...a Cecilia...

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Applications in various domains often lead to high-dimensional data, which put up the challenge of interpreting a huge mass of data, which often consists of millions of measurements. A first step towards addressing this challenge is the use of data reduction techniques, which is essential in the data mining process to reveal natural structures and to identify interesting patterns in the analyzed data.

There are two different approaches to synthesize the two modes (rows and columns) of a high-dimensional data matrix:

1) Asymmetric treatment of the two modes of the data matrix. Here, the two modes assume a different role. The first mode represents objects and is summarized by clustering methods; while the other mode refers to variables and is reduced according to factorial techniques.

2) Symmetric treatment of the two modes of the data matrix. In this case, the two modes have equal role and both are synthesized by clustering techniques. However, we will continue to call these modes as units and variables, even though they will be treated in the same way.

Both approaches can be viewed under a probabilistic or a non probabilistic context.

As far as the asymmetric treatment is concerned, the main task is to partition a given data set into groups on the basis of specified features (variables) so that objects (units, observations) within a group are homogeneous, while objects in separate groups are heterogeneous (Gordon, 1999). However, the researcher could choose to apply factorial reductions of features for two reasons:

1) the number of variables is too large and some of these do not contribute much to identify the clustering structure in the data set;

2) in clustering based on mixture models, if the number of variables is large relative to sample size, it may not be possible to estimate model parameters. In this context, the objects to be clustered are independent realizations on n J-dimensional random vectors (where J is the number of variables). However, some constraints must be imposed on these quantities, since if $n < J$ singular estimates of the component-specific covariance matrices may be obtained.

A frequently used asymmetric approach performs a factorial reduction before using a clustering algorithm. This sequential procedure is often named *tandem analysis* (see for example Chang, 1983). However, the factorial reduction step could remove some information which is relevant for the clustering structure of the data. To overcome this problem, some authors have proposed methods for simultaneous clustering and factorial reduction of the analyzed data (see for example De Soete and Carroll, 1994; Ghahramani and Hinton, 1996; Tipping and Bishop, 1997; Vichi and Kiers, 2001; Rocci and Vichi, 2002).

In the symmetric treatment the task is to capture blocks formed by a subset of units across a subset of variables. In some real fields, such as the text/web mining, microarray data analysis, marketing and preference data analyses, it could be meaningful and informative to cluster both units and variables.

A first approach has been to apply the standard clustering methods to both units and variables successively and independently (see e.g. Tryon, 1939). In this sequential approach, two loss functions are considered, and the final result depends on whether units or variables are classified first.

A more rational procedure is to partition units and variables simultaneously rather than successively (Fisher, 1969). In this case, we have only one loss function to be optimized for both modes of the data matrix.

In the following, the latter approach will be referred to as *double clustering.* The basic idea is to identify *blocks*, i.e., sub-matrices of the observed data matrix, where units and variables (together) specify a *unit cluster* and a *variable cluster.*

Although the double clustering techniques are useful in many application areas, we will focus on a specialized application area which is *microarray expression level modeling.* The past decade has witnessed an explosion of genetic sequence data, culminating with the publication of the draft sequence of the human genome (International human genome sequencing consortium, 2001; Venter et al., 2001). Ten thousands of coding regions of the genetic sequence commonly known as genes have been identified. The gene set of yeast and other simple organisms have been completely characterized, while almost two-thirds of human genes have been identified (International human genome sequencing consortium, 2001). However, sequencing the genome and identifying coding sequences is only a first step in understanding the control and function of an organism at the cellular level. The major challenge is to understand the gene activity, and how different genes do interact. The explosion of data provided by gene microarray technologies has led to an urgent need for novel statistical and computational methods. The process of transforming microarray data into meaningful biological insights is limited by the complexity of corresponding data. One of the most commonly used approaches to analyze microarray data is to apply standard clustering methods. Unfortunately, these are based on assumptions that do not accommodate all the features of gene expression data. Such limitations have motivated the development of several new methods for clustering microarray data.

One of the characteristics of gene expression data is that it is meaningful to cluster both genes and tissue samples (in general conditions). On one hand, co-expressed genes can be grouped in clusters based on their expression patterns by treating, into a clustering context, genes as units and tissue samples as variables (see, e.g., Eisen et al., 1998; Alon et al., 1999). On the other hand, the tissue samples can be partitioned into homogeneous groups, where each group may correspond to some particular macroscopic phenotype, such as cancer types. In this case, tissue samples represents observational units, while genes are variables (see, e.g., Golub et al., 1999; Li et al., 2001).

When clustering genes the problem is that usually co-expressed genes are to be detected only in subsets of samples. In other words, different subsets of variables (samples) are responsible for different co-expressions of genes. Moreover, when clustering samples this situation is even worse. As various phenotypes, e.g. hair color, gender, cancer, etc., are hidden in varying subsets of genes, samples could usually be clustered according to these phenotypes, i.e. in varying subspaces. This belief calls for the simultaneous clustering of genes and tissue samples to capture blocks formed by a subset of genes over a subset of tissue samples (see, e.g., Getz and Domany, 2000; Cheng and Church, 2000 and Madeira and Oliveira, 2004 for a structured overview of simultaneous clustering in microarray data analysis).

The dissertation consists of two parts. In the first part, we briefly remind the K-means method and introduce in details the finite mixture model. Then, we compare two approaches for simultaneous factorial reduction and clustering in a finite mixture context. Moreover, wide space is devoted to some proposals of model-based double clustering methodologies. In the second part, the proposed methods are fitted on simulated and real data to highlight corresponding performance and features.

Methods Chapter 2 introduces two known clustering methodologies: the K-means and the finite mixture model that represent our starting point. The advantages of the finite mixture models in a context of clustering is highlighted; in particular, its power to solve important practical questions in conventional clustering methods (such as how many clusters are there and which similarity measure should be used). Unfortunately, the clustering of high-dimensional data with finite mixture model has some limitations. When the aim is to cluster units on a large number of variables, we face problems in parameter estimation. Chapter 3 describes an attempt to overcome these problems through a factorial reduction step embedded within a standard finite mixture model. Two models are discussed and compared; namely the model proposed by Rocci and Vichi (2002) and the one proposed by Ghahramani and Hinton (1996), successively extended by McLachlan et al. (2000b). The aim is to stress the advantages of using one instead of the other.

In some contexts, factorial reduction methods can have some drawbacks. First, the obtained latent variables often have no intuitive meaning and thus the resulting clusters are hard to interpret. Secondly, if factorial reduction techniques are used, data are clustered only in a particular subspace. In these cases, a potential alternative is to use double clustering methods; that is, methods where the common issue is to identify sub-matrices (blocks) of the observed data matrix, which satisfy specific characteristics of homogeneity. Chapter 4 provides key principles underlying double clustering techniques using both traditional and bioinformatic perspectives. Chapter 5 introduces two new proposals for model-based double clustering: the first is an extension of the double K-means introduced by Vichi (2000) in a probabilistic framework with the aim of defining less arbitrary criterion to select the number of clusters.

The second proposal is a hierarchical extension of standard mixture model; it combines the advantages of allowing for both dependence within clusters and simultaneous clustering of units and variables. The proposed approach is obtained by extending the multilevel latent class model proposed by Vermunt (2003) and Li (2005) to two-way continuous data. Thanks to the hierarchical structure, we may distinguish clusters (2nd level) from components (1st level) giving flexibility to represent extra Gaussian departures. In order to cluster variables, we introduce a binary row stochastic matrix representing variable membership (as in double K-means, Vichi, 2000) through a specific reparameterization of component-specific mean following a path similar to Rocci and Vichi (2002).

Applications The second Part entails applications on both simulated and real data sets. After showing in Chapter 1 the results for both proposed methodologies on simulated data; we describe in details the gene microarray technologies (see Section 7.1). In Section 7.2, the first data set of Bittner et al. (2000) will be described (cDNA microarray). Here, the aim is to define blocks formed by clusters of genes and clusters of tissue samples of cutaneous melanomas. In Sections 7.3, the performance of these methods are illustrated and evaluated.

In Section 7.4, the data set analyzed in Golub et al. (1999) is described. It represents a study on gene expression in two types of acute leukemias: acute lymphoblastic leukemia (ALL) (in turn subdivided into B-cell and T-cell leukemia) and acute myeloid leukemia (AML). In this case, the gene expression levels are measured using Affymetrix high-density oligonucleotide arrays (HU6800chip) to identify genes that distinguish between three known different classes of tissue samples. In Section 7.5, the performance of these methods is illustrated and evaluated. For both methodologies and data sets, the results are encouraging and represent an improvement with respect to known results on the same benchmark data sets. Some conclusions and future research agenda are discussed in the last Chapter.

Part I

METHODS

Chapter 2

Clustering Methods

Clustering methods have been studied for many decades and in many disciplines. They generally aim at identifying subsets (called clusters, groups or classes) of the whole data by measuring proximity or similarity between single units. In this Chapter, we will use *clusters* and *classes* to refer to conceptually meaningful sets of observations that share common features.

Intuitively, units within a group have to be similar (or related) to one another and different from (or unrelated to) the units in other groups. The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better the clustering. The notion of similarity can be expressed in very different ways, according to the purpose of the study, to domain-specific assumptions and to prior knowledge of the problem.

Frequently, the data structure detected by clustering methods can give first insights into the data generating process. Clustering can, therefore, be seen as a useful technique for data mining and knowledge discovery.

The variety of techniques for representing data, the difficulty in establishing similarity between data units, differences in assumptions and contexts in different communities has produced a wide and often confusing range of clustering methods.

First of all, it is important to state the differences between clustering (i.e. *unsupervised classification*) and discriminant analysis (i.e. *supervised classification*). Clustering is usually performed when no information is available about the membership of units to some predefined classes. Thus, clustering is distinguished from pattern recognition or from the areas of statistics known as discriminant analysis and decision analysis, which seek to find rules for classifying units starting from perfect knowledge of pre-classified units.

Within the unsupervised classification framework we can observe two main statistical categories that we will termed as *standard* and *model-based* clustering.

The standard clustering approach is based on intuitively reasonable procedures. One common class of these methods involve *agglomerative hierarchical clustering* (Johnson, 1967), in which two groups are merged iteratively to optimize some criteria. In contrast, the *divisive hierarchical clustering* starts with all units in one cluster and subdivide them into smaller groups. Divisive methods have rarely been applied.

Another widely used class of standard methods is based on *iterative relocation* (partitioning), where data points are moved from one group to another until there is no further improvement in some criterion function (loss function). Iterative relocation with a LS criterion is often called *K-means* clustering (McQueen, 1967). This algorithm belongs to the class of *exclusive clustering algorithms*, where data are grouped in an exclusive way, so that if a unit belongs to a certain cluster then it could not be included in another cluster.

Although there have been considerable improvements in these standard approaches, this class of methods does not solve basic practical questions arising in cluster analysis, such as how many clusters there are and which clustering method should be used.

In contrast, cluster analysis can also be based on probability models (see Bock, 1996 and 1998; for survey). So called *Model-based clustering* approaches describe data as being generated by some probabilistic process, and the goal of clustering is to recover the parameters of that process. The clustering algorithm fits a probabilistic model to the data (usually by maximum likelihood, with an overfitting penalty), and the quality of the fitted model can be evaluated by measuring the likelihood corresponding to a separate test data set.

These approaches have provided further insight into those clustering methods which can be expected to work well (i.e., the data conform to the model), and have led to the development of new clustering methods. It has also been shown that some of the most popular standard clustering methods can be viewed as estimation methods for certain known probability models. For example, K-means clustering algorithm has been shown to be closely related to model-based clustering using the equal volume spherical model, as computed by the EM algorithm (Celeux and Govaert, 1992). In other words, K-means is an approximate estimation method for a parsimonious model based on simple independent Gaussians.

Model-based clustering techniques provide a statistical approach to solving important practical questions that arise in applying clustering methods. An advantage of using a statistical model is that choosing both the cluster criterion and the number of clusters is less arbitrary since the approach includes rigorous formal criteria based on the log-likelihood function penalized with model parameters. In fact, the basic idea of model-based clustering is to approximate the data density by a mixture model, typically a mixture of Gaussians. The number of distinct groups in the data is then taken to be the number of mixture components, and the observations are partitioned into clusters using Bayes' rule. A drawback of this approach is that if the groups in the data are well separated and look Gaussian, then model-based clustering will produce clusters that indeed tend to be "distinct" in the most common sense of the word (contiguous, densely populated areas of feature space, separated by contiguous, relatively empty regions). If the groups are not Gaussian or if the covariance structure of the mixture model is incorrect, this correspondence may break down; an isolated group with a non-elliptical distribution, for example, may be modeled by several mixture components, and the corresponding clusters will no longer be well separated.

Another advantage of the model-based approach is that no scaling of the observed variables is needed. For instance, when working with Gaussian distributions with unknown variances, the results will be the same irrespective of whether the variables are normalized or not. This is not so for standard cluster methods like K-means, where scaling is always an issue. Other advantages are that it is relatively easy to deal with variables of mixed measurement levels (different scale types). Such a use of model-based clustering techniques has been referred to as the mixture likelihood approach, latent class analysis, mixture model clustering, Bayesian Classification and unsupervised learning. In the marketing research field, model-based clustering is sometimes referred to as latent discriminant analysis (Dillon and Mulani, 1989) because of the similarity to the statistical methods used in discriminant analysis as well as in logistic regression. However, an important difference is that in discriminant (and logistic regression) group membership is assumed to be known while in model-based clustering it is latent and, therefore, unobservable.

In this Chapter, a review of two methods, namely the K-means and mixture model [1] clustering, belonging to standard and model-based approaches respectively, will be given. These methods represents probably the standard benchmarks in both fields, and have been longely debated and widely extended to deal with a variety of empirical applications. In particular, Section 2.1 describes the standard K-means clustering while Section 2.2 describes the finite mixture model proposed by Wolfe (1963) and further extended by many other authors (see McLachlan and Basford, 1988; Banfield and Raftery, 1993; McLachlan et al., 1999). We are mainly concerned with the case where each mixture component density is a multivariate Gaussian, a model that has gained considerable success in a number of applications (see e.g. Murtagh and Raftery, 1984; Celeux and Govaert, 1995; Dasgupta and Raftery, 1998). In Section 2.3 we show that some standard clustering methods can be analyzed under a probabilistic framework.

The last three Sections deal with some topics in the context of model-based approach. In particular, Section 2.4 introduces the specific problem of choosing initial values for the EM algorithm while in Section 2.5 some useful extensions are showed. The last Section examines different approaches to model selection.

2.1 The K-means algorithm

The K-means method (McQueen, 1967) is one of the most simple unsupervised learning algorithms that discovers groups in the data. The procedure, which is shown in Table 2.1, follows an easy iterative way to classify a given data set in a certain number, say K, of clusters. The main idea is to define K

[1] It has to be noted that the advantages in using the model-based approach is not just to cluster analysis, but also to some other basic problems of discriminant analysis and multivariate density estimation (see McLachlan and Peel, 2000a).

centroids, one for each cluster. It has to be noted that the choice of different locations for centroids causes different results. So, the better choice is to place them as much as possible far away from each other. The second step is to take each point belonging to the data set and allocate it to the nearest centroid. When no points are pending, the second step is completed and an early grouping is done. At this point K new centroids are calculated as barycenters of the groups resulting from the previous step (see Figure 2.1). There, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the K centroids change their location step by step until no more changes are necessary. In other words, centroids do not move any more. In order to reduce the computational complexity, the previous stopping rule can be replaced by a less restrictive rule such as the one that stops the procedure if:

- the distance between each centroid in the current iteration and the corresponding centroid in the previous iteration is not higher than a fixed threshold;

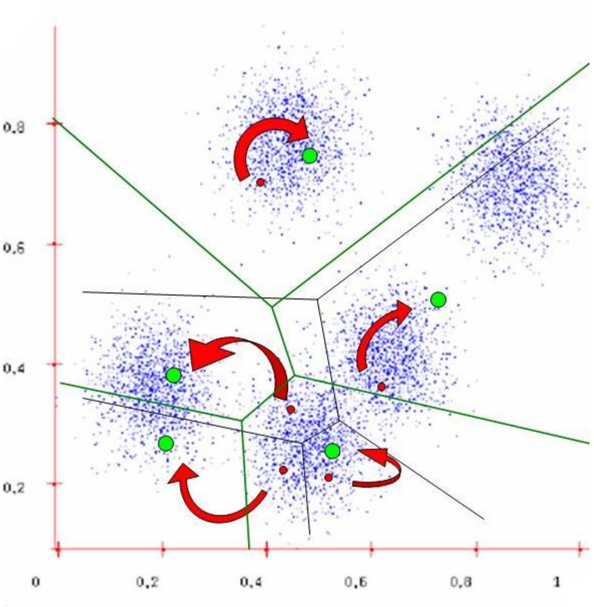
- the maximum number of iterations is achieved.

The metric used to calculate these distances is usually the Euclidean one, since it guarantees the convergence of the iterative procedure (Anderberg, 1973). Thus, at the t-th iteration the distance between the i-th unit and the k-th centroid ($i = 1, ..., n$; $k = 1, ..., K$) is given by:

$$d(y_i, \bar{y}_k^{(t)}) = \sqrt{\sum_{j=1}^{J} (y_{ij} - \bar{y}_{kj})^2} \tag{2.1}$$

where y_{ij} is the value of the j-th variable on the i-th unit, $\bar{y}_{kj}$ represents the value of the j-th variable on the k-th centroid.

Figure 2.1: K-means: new nearest cluster centroids



It is clear from (2.1) that he K-means algorithm aims at minimizing a squared loss function, W, which is given by:

$$W = \sum_{k=1}^{K} W_k = \sum_{k=1}^{K} [\sum_{i=1}^{n_k} \sum_{j=1}^{J} (y_{ij} - \bar{y}_{kj})^2], \tag{2.2}$$

where the quantity $[\sum_{j=1}^{J} \sum_{i=1}^{n_k} (y_{ij} - \bar{y}_{kj})^2]$ indicates the deviance within the k-th group (of size n_k).

Although it can be proved that the procedure will always converge, the K-means algorithm does not necessarily find the global minimum function of the loss. The algorithm is also significantly sensitive to the initial choice of cluster centers. Thus, the K-means algorithm has to be run several times to reduce this effect.

There are additional steps that can be found in variations of the standard K-means algorithm described above. Quackenbush (2001) discusses a few optional steps and variants of this algorithm. Despite K-means algo-

Table 2.1: K-means algorithm

1- Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2- Assign each object to the group corresponding to the closest centroid.
3- When all objects have been assigned, recalculate the positions of the K centroids.
4- Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups; the loss function to be minimized can be calculated.

rithm is very easy to implement, it provides no direct heuristic measure to guide the analyst as to the number of clusters within the data. Tamayo et al. (1999) explain that "K-means clustering is a completely unstructured approach, which proceeds in a entirely local fashion and produces an unorganized collection of clusters that is not conductive to interpretation".

It can be stressed that K-means method can be viewed as a particular case of mixture models (Section 2.3).

2.2 Finite mixture models

Let $\mathbf{Y}_i$ be a J-dimensional random vector and $\mathbf{y}_i$ its generic realization ($i = 1, ..., n$). Let $\mathbf{Y} = (\mathbf{Y}_1, ..., \mathbf{Y}_n)'$ be the data matrix, where the generic element y_{ij} represents the value of the j-th variable on the i-th unit ($i = 1, ..., n$; $j = 1, ..., J$). A mixture model assumes that each observation $\mathbf{y}_i$ ($i = 1, ..., n$) is drawn from a mixture of K groups (corresponding to mixture component densities) in some unknown mixing proportions $\pi_1, ..., \pi_K$. In other words, $\mathbf{y}_i$ has density function defined by:

$$f(\mathbf{y}_i; \boldsymbol{\phi}) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{y}_i; \boldsymbol{\theta}_k) \tag{2.3}$$

where $f_k(\mathbf{y}_i; \boldsymbol{\theta}_k)$ denotes the k-th component density with parameter vector $\boldsymbol{\theta}_k$, the π_k's represent mixing weights with $\pi_k \geq 0$, $\sum_{k=1}^{K} \pi_k = 1$, while $\boldsymbol{\phi} = (\pi_1, ..., \pi_{K-1}, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K)$ denote the *complete* parameter vector.

If observations within the k-th component follow a J-variate Gaussian density function with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, we have:

$$f(\mathbf{y}_i; \boldsymbol{\phi}) = \sum_{k=1}^{K} \pi_k \varphi(\mathbf{y}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) =$$

$$= \sum_{k=1}^{K} \pi_k \left[\frac{1}{\sqrt{(2\pi)^J}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_k) \right\} \right]. \tag{2.4}$$

Thus, the clusters associated to the mixture components are ellipsoidal, centered at $\boldsymbol{\mu}_k$; the covariance matrices $\boldsymbol{\Sigma}_k$ determine the geometric features of these ellipses.

In order to characterize the mixture model, i.e. to estimate its parameters, several approaches may be considered. As exposed by McLachlan and Peel (2000a), such approaches include graphical methods, methods of moments, minimum-distance methods, maximum likelihood and Bayesian methods. However, the maximum likelihood (ML) framework is the most commonly used approach to fitting mixture models. The optimization of the likelihood function of $\boldsymbol{\phi}$ given the data $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_n)$ is analytically intractable, and we must resort to more elaborate techniques. The Expectation-Maximization (EM) algorithm (Dempster et al., 1977) is one such technique. It is a fixed-point iterative method that locally maximizes the likelihood function in an efficient way. In particular, the EM algorithm considerably simplifies the ML approach to parameter estimation by assuming the existence of missing data and posing the mixture model into an incomplete-data problem. Let us define $\mathbf{z}_i = (z_{i1}, ..., z_{iK})$ with $z_{ik} = 1$ or 0 according to whether $\mathbf{y}_i$ is drawn from k-th mixture component or not.

This algorithm assumes that the observations $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_n)$ are incomplete since we have no available information on indicator vectors or labels $\mathbf{z} = (\mathbf{z}_1, ...\mathbf{z}_n)$.

It is worth noting that if the sample $\mathbf{z}$ is known we are in a *discriminant analysis* context where the problem is essentially to predict an indicator vector $\mathbf{z}_{n+1}$ from a new observed data vector $\mathbf{y}_{n+1}$. On the other hand, if the sample $\mathbf{z}$ is unknown we are in a *density estimation* or *cluster analysis* context. Here, we consider only the latter case; we assume the vectors $(\mathbf{z}_1, ..., \mathbf{z}_n)$ are drawn from a Multinomial distribution with prior probabilities π_k. The log-likelihood function for the complete-data $(\mathbf{y}, \mathbf{z})$ is defined as

$$\log L_c(\boldsymbol{\phi}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log\big[\pi_k f(\mathbf{y}_i; \boldsymbol{\theta}_k)\big] = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \Big\{\log(\pi_k) + \log\big[f(\mathbf{y}_i; \boldsymbol{\theta}_k)\big]\Big\}. \tag{2.5}$$

The Expectation Maximization algorithm is made up by two steps: in the $(h+1)$-th iteration of the E-step, we compute the expected value of the complete log-likelihood function, conditional on the observed data and the current parameter estimate $\boldsymbol{\phi}^{(h)}$, say $Q(\boldsymbol{\phi}, \boldsymbol{\phi}^{(h)})$. Since the expected value is linear in the missing variables, the E-step reduces to the computation of terms.

$$w_{ik}^{(h)} = \frac{\hat{\pi}_k^{(h)} \varphi(\mathbf{y}_i; \hat{\boldsymbol{\mu}}_k^{(h)}, \hat{\boldsymbol{\Sigma}}_k^{(h)})}{\sum_{k=1}^{K} \hat{\pi}_k^{(h)} \varphi(\mathbf{y}_i; \hat{\boldsymbol{\mu}}_k^{(h)}, \hat{\boldsymbol{\Sigma}}_k^{(h)})} = Pr(z_{ik} = 1 | \mathbf{y}; \boldsymbol{\phi}^{(h)}). \tag{2.6}$$

The $(h+1)$-th iteration of the M-step, instead, updates parameter estimates by maximizing the expected value of the complete log-likelihood function given the weights $w_{ik}^{(h)}$. The estimates, when Gaussian component-specific densities are used, have a simple closed form involving the data and the w_{ik} calculated in the E step,

$$\hat{\pi}_k^{(h+1)} = \frac{\sum_{i=1}^{n} w_{ik}^{(h)}}{n} \tag{2.7}$$

$$\hat{\boldsymbol{\mu}}_k^{(h+1)} = \frac{\sum_{i=1}^{n} \mathbf{y}_i w_{ik}^{(h)}}{\sum_{i=1}^{n} w_{ik}^{(h)}}. \tag{2.8}$$

Computation of the covariance matrices estimates depends on the adopted reparameterization. For Gaussian models Banfield and Raftery (1993) and Celeux and Govaert (1995) considered a reparameterization of the component-specific covariance matrix, $\boldsymbol{\Sigma}_k$, in terms of its eigenvalue decomposition:

$$\boldsymbol{\Sigma}_k = \lambda_k \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k', \tag{2.9}$$

where $\lambda_k = |\boldsymbol{\Sigma}_k|^{1/J}$, $\mathbf{D}_k$ is the orthogonal matrix of eigenvectors of $\boldsymbol{\Sigma}_k$ and $\mathbf{A}_k$ is a diagonal matrix, with $|\mathbf{A}_k| = 1$, where diagonal entries correspond to the normalized eigenvalues of $\boldsymbol{\Sigma}_k$ in decreasing order. The matrix $\mathbf{D}_k$ determines the orientation of the mixture component, $\mathbf{A}_k$ determines the shape of the mixture component, and λ_k determines its volume.

Allowing some but not all of the parameters in equation (2.9) to vary results in a set of 14 specific models. Here, the 14 models will be classified into three main families: *general*, *diagonal* and *spherical*. In detail, the general family allows volumes, shapes and orientations of clusters to vary or to be equal between clusters. Variations on the parameters λ_k, $\mathbf{D}_k$, and $\mathbf{A}_k$ $(k = 1, ..., K)$ lead to 8 general models. The diagonal family assumes that the covariance matrix, $\mathbf{\Sigma}_k$, is diagonal. This means that the matrices $\mathbf{D}_k$ are permutation matrices. By writing $\mathbf{\Sigma}_k = \lambda_k \mathbf{B}_k$, where $\mathbf{B}_k$ is a diagonal matrix with $|\mathbf{B}_k| = 1$, we have 4 additional models. The last family of models assumes spherical shapes constraining $\mathbf{A}_k$ to be equal to a identity matrix I. In such a case, 2 models are possible. Table 2.2 summarizes some features of the 14 models. The first column specifies the model; the second column gives the number of parameters to be estimated and the third column indicates if the M step can be achieved with closed form formulas (CF) or if we have to use an iterative procedure (IP). Last column shows the kind of function to be minimized when the M step has a closed form. As it can be seen, the most general form of the covariance matrix estimate is:

$$\hat{\mathbf{\Sigma}}_k^{(h+1)} = \frac{\sum_{i=1}^{n} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_k^{(h+1)})(\mathbf{y}_i - \hat{\boldsymbol{\mu}}_k^{(h+1)})' w_{ik}^{(h)}}{\sum_{i=1}^{n} w_{ik}^{(h)}}. \tag{2.10}$$

As a by product, the EM algorithm provides a (fuzzy) posterior matrix of component membership. Therefore, we can cluster objects $i = 1, ..., n$ according to posterior probabilities of component membership, using a Maximum a Posteriori (MaP) approach, that is:

$$\widehat{z}_{ik} = \begin{cases} 1 & \text{if } k = \text{argmax}_{k=1,...,K} w_{ik} \\ 0 & \text{otherwise} \end{cases}. \tag{2.11}$$

Many authors (for example Wu, 1983; Boyles, 1983) have shown that under general regularity conditions, the solution will converge to a local maximum of the likelihood function. Moreover, the EM algorithm typically gives good results if data conform reasonably well to the model and the iterations are started at reasonable values. However, the EM algorithm for multivariate Gaussian mixtures breaks down when the covariance associated with one or more components is singular or nearly singular. It may either fail or give inaccurate results if one or more clusters contain only few observations (which can happen if there are too many components in mixture), or if the observations they contain are concentrated close to a linear subspace of reduced dimension.

Table 2.2: Some characteristics of the 14 models

model	number of parameters	M step	criteria
$\lambda\mathbf{DAD}'$	$\alpha+\beta$	CF	$\lvert\mathbf{V}\rvert$
$\lambda_k\mathbf{DAD}'$	$\alpha+\beta+K-1$	IP	-
$\lambda\mathbf{DA}_k\mathbf{D}'$	$\alpha+\beta+(K-1)(J-1)$	IP	-
$\lambda_k\mathbf{DA}_k\mathbf{D}'$	$\alpha+\beta+(K-1)J$	IP	-
$\lambda\mathbf{D}_k\mathbf{AD}_k'$	$\alpha+K\beta-(K-1)J$	CF	$\lvert\mathbf{\Sigma}_k\Omega_k\rvert$
$\lambda_k\mathbf{D}_k\mathbf{AD}_k'$	$\alpha+K\beta-(K-1)(J-1)$	IP	-
$\lambda\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k'$	$\alpha+K\beta-(K-1)$	CF	$\mathbf{\Sigma}_k\lvert\mathbf{V}_k\rvert^{1/J}$
$\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k'$	$\alpha+K$	CF	$\mathbf{\Sigma}_k n_k\ln(\frac{\lvert\mathbf{V}_k\rvert}{n_k})$
$\lambda\mathbf{B}$	$\alpha+J$	CF	$diag\lvert(\mathbf{V})\rvert$
$\lambda_k\mathbf{B}$	$\alpha+J+K-1$	IP	-
$\lambda\mathbf{B}_k$	$\alpha+JK-K+1$	CF	$\mathbf{\Sigma}_k\lvert diag(\mathbf{V}_k)\rvert^{1/J}$
$\lambda_k\mathbf{B}$	$\alpha+KJ$	CF	$\mathbf{\Sigma}_k n_k\ln(diag(\frac{\lvert\mathbf{V}_k\rvert}{n_k}))$
$\lambda\mathbf{I}$	$\alpha+1$	CF	$tr(\mathbf{V})$
$\lambda_k\mathbf{I}_k$	$\alpha+J$	CF	$\mathbf{\Sigma}_k n_k\ln(tr(\frac{\mathbf{V}_k}{n_k}))$

We have $\alpha = KJ+K-1$ in the case of free weights and $\alpha = KJ$ in the case of equal weights, and $\beta = \frac{J(J+1)}{2}$. Further, we have $\mathbf{V}_k = \sum_{i=1}^{n} w_{ik}(\mathbf{y}_i - \boldsymbol{\mu}_k)(\mathbf{y}_i - \boldsymbol{\mu}_k)'$; $\mathbf{V} = \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik}(\mathbf{y}_i - \boldsymbol{\mu}_k)(\mathbf{y}_i - \boldsymbol{\mu}_k)'$ and Ω_k is the diagonal matrix with the eigenvalues of $\mathbf{V}_k$ in decreasing order

2.3 Standard clustering methods and probabilistic framework

In a mixture model context, the log-likelihood function to be maximized could be written as follows:

$$C(P, \boldsymbol{\phi}) = \sum_{k=1}^{K} \sum_{\mathbf{y}_i \in P_k} \log\bigl[\pi_k f(\mathbf{y}_i; \boldsymbol{\theta}_k)\bigr],$$

where $P = (P_1, ..., P_K)$ is a partition of the sample $\mathbf{y}_1, ..., \mathbf{y}_n$ associated to the indicator vectors $\mathbf{z}_1, ..., \mathbf{z}_n$, $\mathbf{y}_i$ a J-dimensional vector and $\boldsymbol{\phi} = \{\pi_1, ..., \pi_k; \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K\}$ the set of parameters to be estimated ($i = 1, ..., n$; $k = 1, ..., K$).

Moreover, let us remind that the most popular clustering criterion, the so-called variance criterion, is defined by

$$W(P) = \sum_{k=1}^{K} W(P_k) = \sum_{k=1}^{K} \sum_{\mathbf{y}_i \in P_k} \parallel \mathbf{y}_i - \bar{\mathbf{y}}_k \parallel^2$$

where $\bar{\mathbf{y}}_k$ is the center of the cluster P_k ($k = 1, ..., K$).

The following proposition highlights the link between $C(P, \boldsymbol{\phi})$ and $W(P)$ criteria.

Proposition. Celeux and Govaert (1992) show that maximizing the C criterion for a Gaussian mixture with equal weights and a common covariance matrix of the form $\sigma^2\mathbf{I}$ (σ^2 unknown) is equivalent to minimize the variance criterion W.

Proof. In the Gaussian case, we have $\boldsymbol{\theta}_k = (\mu_k, \sigma^2)$ and $\pi_k = 1/K$ ($k = 1, ..., K$). For a fixed partition $P = (P_1, ..., P_K)$, it can be easily proved that the maximum likelihood estimate of μ_k is the center of cluster P_k. In these conditions, C can be written as

$$C(P, \boldsymbol{\phi}) = -\frac{1}{\sigma^2} W(P) - nJ \log(\sigma^2) + A$$

where A denotes a constant. It is clear to see that the estimate of σ^2 optimizing C is $W(P)/nJ$.

From this proposition, it is straightforward to see that the EM method is a natural extension of K-means method. Whereas a hard membership is adopted in the K-means algorithm (i.e. a data pattern is assigned to one cluster only), a soft membership is allowed in the EM algorithm (i.e. the membership of each data pattern can be distributed over multiple clusters).

The key difference of K-means and EM algorithm lies in the fact that the latter was originally proposed for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. In particular, it is frequently used as a parametric method of clustering by assuming a mixture model for the data with different distributions for the clusters and unknown mixing proportions. Here the missing data are the true cluster memberships of the observations and the number of clusters, K, can be estimated using some model selection criteria.

On other hand, the K-means algorithm is a nonparametric method for clustering as it does not assume any probability model for the data. Given a fixed number of clusters, it determines an assignment of the data vectors (observations) to the clusters so as to minimize the total of the squared distances between the observations assigned to the same cluster and summed over all clusters.

This difference is highlighted by the fact that the loss function $W(P)$ monotonously decreases with the increase of K, while $C(P, \boldsymbol{\phi})$ may not increase with the increase of K.

2.4 Choice of the initial values for the EM algorithm

The first step of the EM algorithm requires the calculation of the expected value of the log-likelihood function of the complete data, conditional on the observed data, $\mathbf{y}$, and the initial vector of parameter estimates $\hat{\boldsymbol{\phi}}_0$. The choice of this vector meaningfully influences the algorithm's speed and its convergence. In fact, a slow convergence of the EM algorithm could due to a "bad" choice of the vector $\hat{\boldsymbol{\phi}}_0$. If the likelihood function is not limited at the bounds of the parameter space, the sequence of estimates $\{\hat{\boldsymbol{\phi}}_h\}$ given by the algorithm could diverge if $\hat{\boldsymbol{\phi}}_0$ is chosen too near to the bounds.

Another problem connected to mixture models regards the case when the likelihood equation has multiple roots that correspond to local maxima; in this case, the EM algorithm could be applied choosing among a vast set of initial values. Without other information, an appropriate choice among the roots of the likelihood equation is the greatest local maximum, although this choice does not guarantee a valid and asymptotically efficient sequence of roots (Lehmann, 1980).

In the context of mixture models with independent data, the E-step is reduced to update the posterior probability of an unit to belong to a mixture component. Therefore, an alternative approach can be to specify a value $\mathbf{w}_i^{(0)}$ to $\mathbf{w}_i$ for $i = 1, ..., n$, where

$$\mathbf{w}_i = (w_{i1}, ..., w_{iK})'$$

is the vector for the i-th unit containing the K posterior probabilities to belong to a mixture component. An usual choice is $\mathbf{w}_i^{(0)} = \mathbf{z}_i^{(0)}$, for $i = 1, ..., n$, where $\mathbf{z}^{(0)} = (\mathbf{z}_1^{(0)}, ..., \mathbf{z}_n^{(0)})'$ defines a initial partition of data in K

groups. For example, an ad-hoc way to provide an initial partition in the case of a two Gaussian component mixture model with same covariance matrix, could be to plot data in order to select two of the J variables, taking out a straight line dividing bivariate data into two groups with a Gaussian form. For high-dimensionality data, an initial value $\mathbf{z}^{(0)}$ for $\mathbf{z}$ can be obtained using as starting points the solution of some clustering algorithms, such as the K-means or, if sample size is not too large, some hierarchical procedures.

Another way to specify an initial partition $\mathbf{z}^{(0)}$ of the data is to randomly divide data in K initial values of the components. In other words, for each observation $\mathbf{y}_i$, we generate an integer included in $[1, ..., K]$. If the random integer is equal to t, we put $\mathbf{z}_{it}^{(0)} = 1$, $t = 1, ..., K$.

An alternative method to choose an initial value, in the case of J-variate Gaussian models with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, is to randomly generate $\boldsymbol{\mu}_k^0$ as:

$$\boldsymbol{\mu}_1^0, ..., \boldsymbol{\mu}_K^0 \sim N(\overline{\mathbf{y}}, \mathbf{V})$$

where $\overline{\mathbf{y}}$ and $\mathbf{V}$ correspond to the sample mean and the sample covariance matrix respectively. However, using this method, we have more variability in the initial values $\boldsymbol{\mu}_k^0$ than using a random partitioning of the data into K groups. We can specify the covariance matrices and the mixing proportions respectively as:

$$\boldsymbol{\Sigma}_k^0 = \mathbf{V}$$

and

$$\pi_k^0 = 1/K$$

for $k = 1, ..., K$.

As shown in McLachlan and Basford (1988), a key problem in mixture models is how to obtain a good estimate of the mixing weights. For univariate mixtures, Fowlkes (1979) suggests to determine the flex point in the quantile-quantile plots to estimate the mixing weights in the sub-populations. The remaining parameters can then be estimated through the sample once it is partitioned according to these estimates. Ichihashai, Honda e Tani (2000) show that the EM algorithm for Gaussian mixture models can be produced by a modified version of the K-means fuzzy method, in which the Kullback-Leibler information is introduced as term of "regularization" instead of the entropy, that is, the loss function is written as:

$$J_{\lambda\tau} = \sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})d_{ik}^{2} + \lambda\sum_{k=1}^{K}\sum_{i=1}^{n}u_{ik}\log(\frac{u_{ik}}{\pi_k}) - \sum_{i=1}^{n}\eta_i(\sum_{k=1}^{K}u_{ik}-1) - \tau(\sum_{k=1}^{K}\pi_k-1) \tag{2.12}$$

where the membership function, u_{ik}, is equal to the posterior probability that $\mathbf{y}_i$ belongs to k-th component w_{ik}, d_{ik} is the distance between $\mathbf{y}_i$ and the k-th prototype cluster, λ is a positive parameter, π_k is the prior probability that the unit belongs to the k-th mixture component, η_i and τ are penalization terms. The second term of (2.12) represents the Kullback-Leibler distance, which obtains its minimum if $u_{ik} = \pi_k$, $\forall\ k$. The greater is λ, the more u_{ik} tend to π_k, $i = 1, ..., n$; $k = 1, ..., K$.

2.5 Variants of the EM algorithm

As it has been shown, the EM algorithm is an iterative algorithm; each iteration is based upon two steps, the Expectation Step (E-step) and the Maximization Step (M-step). A brief history of the EM algorithm can be found in McLachlan and Krishnan (1997). The name EM algorithm was coined by Dempster et al. (1977), who synthesized earlier formulations of this algorithm which was already used in many particular cases; they presented a general formulation of the method for finding MLE in a variety of problems and provided a series of problems where this method could be profitably applied. Since then, the EM algorithm has been applied in a large variety of statistical problems such as resolution of mixtures, multi-way contingency tables, variance component estimation, factor analysis, as well as in specialized applications in genetics, medical imaging, and neural networks.

The EM has gained popularity since it is numerically stable in each iteration under fairly general conditions, and has good properties with respect to global convergence. Moreover, it is easily implemented, analytically and computationally. By looking at the monotone increase in likelihood function over iterations, it is easy to monitor convergence and programming errors (McLachlan and Krishnan, 1997). The cost per iteration is generally low, which can offset the larger number of iterations needed for the EM algorithm compared to other competing procedures.

However, the EM algorithm is sometimes very slow to converge and in some problems the E or M steps may be analytically intractable. The next sections review some modifications and extensions of the EM algorithm. In particular, in Section 2.5.1, we focus on the GEM and its subclasses (ECM, ECME, an extension of ECM, AECM and MCEM). In Section 2.5.2 and 2.5.3, we describe in details two further algorithms, the so-called CEM and SEM (stochastic version of CEM) algorithms.

2.5.1 GEM-type algorithms

In the generalized EM algorithm (GEM), defined by Dempster et al. (1977), the expectation of log-likelihood function is is increased (instead of maximized) at each M-step. The M-step requires $\boldsymbol{\phi}_{h+1}$ to increase the expected value of the complete log-likelihood function rather than to maximize it over all $\boldsymbol{\phi} \in \Omega$. Problems arise since in general GEM does not appropriately converge without further specifications of the process of increasing the expectation of log-likelihood function.

Meng and Rubin (1993) proposed generalized EM algorithm which they call the ECM (Expectation, Conditional Maximization) algorithm. The ECM typically converges more slowly than the EM algorithm in terms of the number of iterations, but can be faster with respect to CPU time. Moreover, it preserves the appealing convergence properties of the EM algorithm, such as its monotone convergence. Briefly, the ECM algorithm modifies the EM algorithm by replacing its M-step, which maximizes the current expected complete log-likelihood function over the entire parameter space $\boldsymbol{\Phi}$, by a sequence of conditional maximization steps (indexed by $s = 1, ..., S$), each of which maximizes the expected complete log-likelihood function over a function of $\boldsymbol{\phi}$, say $\boldsymbol{\phi}_s$, conditional on $\overline{\boldsymbol{\phi}_s}$, being fixed at previously estimated values. If ($\boldsymbol{\phi}_1$,...,$\boldsymbol{\phi}_S$) span the parameter space of $\boldsymbol{\phi}$, the ECM algorithm will converge in the same way as the EM does to the ML estimate. Certain restrictions are needed to ensure that the resulting maximum is an unconstrained maximum of the observed likelihood. A necessary condition is that the set of constraint functions is *space filling* (see Wu, 1983).

In many situations, the computation of the E-step may be cheaper than the computation of the CM-steps. Thus, we can think to perform an E-step before each CM-step or before a few selected CM-steps. Here, we consider the case where we perform an E-step before each CM-steps: a cycle is defined to be one E-step followed by one CM-step. Meng and Rubin (1993) called the corresponding algorithm a multicycle ECM; the likelihood function is monotonically increased after each cycle, and hence, after each iteration. A disadvantage of using a multicycle ECM algorithm is that the extra computations at each iteration lead to slow convergence, slower than the ECM.

In the factor analysis context, Liu and Rubin (1994) proposed the ECME (Expectation, Conditional Maximization either) algorithm, which is an extension of the ECM, but with a rate of convergence which is substantially faster than either EM or ECM, while retaining the stable monotone convergence to an ML estimate of the EM algorithm. This increased rate of monotone convergence makes it easier to judge convergence, and total computation time can be lower than with EM.

ECME (Expectation, Conditional Maximization Either) replaces each or more of CM steps with a step that conditionally maximizes the actual likelihood function over $\boldsymbol{\phi}_s$ rather than the expected complete data log-likelihood as with ECM. Typically, the conditional maximization of the actual likelihood over $\boldsymbol{\phi}_s$ is more difficult than the conditional maximization of the expected complete-data log-likelihood over $\boldsymbol{\phi}_s$. Thus, ECME is typically more tedious to implement than ECM, and some of the steps are computationally more expensive. Its features are, however, an increased rate of convergence, and an increased ability to assess convergence and decreased total computer time, both obtained without loosing the monotone increase in likelihood and simple implementation.

Meng and Van Dyk (1997) proposed an extension of ECM algorithm called the AECM (Alternating ECM) algorithm, where the specification of the complete-data is allowed to be different on each CM-step. In other words, the complete-data need not be the same at each CM-step. Thus, the ECME can be considered as a special case of the AECM algorithm, where the complete-data can be specified as the observed data on a CM-step. On the other side, the AECM algorithm can be viewed as a combination of the ECME algorithm and the Space-Alternating Generalized EM (SAGE) algorithm proposed by Fessler and Hero (1994). The Space-Alternating Generalized EM (SAGE) algorithm has been developed without knowledge of ECM and ECME algorithms and performs only one EM iteration on a given CM-step.

ECME itself can be embedded in a more general AECM algorithm, which is closely related to multicycle ECM. Another advantage of using these EM-type algorithms is that large sample standard errors (see warnings in Rubin and Thayer, 1983, concerning their inferential use) can be obtained numerically using only the code for EM (Meng and Rubin, 1991) or ECM (Van Dyk, Meng and Rubin, 1995).

In Section 2.5.3, we show a modified version of the EM algorithm in the context of computing the MLE for the finite mixture models; it is called the Stochastic EM algorithm, and has been proposed by Celeux and Diebolt (1985) before the appearance of the MCEM (Monte Carlo EM) algorithm proposed by Wei and Tanner (1990); it is the same as the MCEM algorithm when M=1.

2.5.2 The CEM algorithm

A variant of the EM is the CEM algorithm (Celeux and Govaert, 1992). This algorithm incorporates a classification step between the E and the M step of an EM algorithm. Starting from an initial parameter vector $\boldsymbol{\phi}_0$, an iteration of the CEM consists of three steps.

- E-step: The conditional probabilities w_{ik}, $i = 1, ..., n$ and $k = 1, ..., K$, are calculated as in the standard E-step.

- C-step: A partition $P = (P_1, ..., P_K)$ of $(\mathbf{y}_1,...,\mathbf{y}_n)$ is designed by assigning each unit to the component maximizing the conditional probability

w_{ik}, $(k = 1, ..., K)$, i.e. adopting a MAP approach.

- M-step: The ML estimate of $\boldsymbol{\phi}$ is updated using the cluster P_k as subsample $(k = 1, ..., K)$ of the k-th mixture component.

CEM maximizes the complete data log-likelihood where the missing component indicator vector of each sample point is included in the data set. As a consequence, CEM is not expected to converge to the ML estimate of $\boldsymbol{\phi}$ and may yield inconsistent estimates of the parameters especially when the mixture components are overlapping or are in disparate proportions (McLachlan and Peel, 2000a).

From a practical point of view, the solution provided by the CEM algorithm does depend upon its initial value, especially in the case in which the clusters are not well separated. Usually, to overcome this problem, the algorithm is repeated several times from different initial values and the clustering which provides the greatest value of the complete log-likelihood function is selected. Obviously, if most of the CEM runs lead to the same clustering, we can be confident that the global optimum has been achieved. In the next Section, we present a stochastic version of the CEM algorithm which has been designed to give an answer to its dependence on initial values.

2.5.3 The SEM algorithm

The SEM algorithm has been proposed by Celeux and Diebolt (1985) to identify the parameters of a mixture model as an alternative to the EM algorithm. It gives an answer to the well known limitations of the EM (strong dependence on initial values, slow convergence, etc) which can occur when the mixture components are not well separated. The SEM algorithm is a stochastic version of the EM incorporating a restoration of the unknown component labels $\mathbf{z}_i$, $i = 1, ..., n$ between the E and M steps, by drawing from their current conditional distribution. Starting from an initial parameter $\boldsymbol{\phi}_0$, an iteration of the SEM algorithm consists of the three following steps.

- E-step: The conditional probabilities w_{ik}, $i = 1, ..., n$ and $k = 1, ..., K$, are calculated as in the standard E-step.

- S-step: A partition $P = (P_1, ..., P_K)$ of $(\mathbf{y}_1,...,\mathbf{y}_n)$ is designed by drawing the component indicator $\mathbf{z}_i$ from a Multinomial distribution with probabilities w_{ik}, $(k = 1, ..., K)$.

- M-step: The ML estimate of $\boldsymbol{\phi}$ is updated using the cluster P_k as subsample $(k = 1, ..., K)$ of the k-th mixture component.

It is clear that the SEM algorithm can be thought of as a stochastic version of the CEM as well as of the EM; in fact, it appears to be a natural stochastic version of both algorithms though they are designed to optimize different criteria: the log-likelihood function for the EM algorithm and the classification log-likelihood function for CEM. Moreover, the two algorithms lead to different partitions. The convex hulls of the clusters generated by the CEM are disjoint whereas the clusters generated by the SEM are generally overlapping. From this point of view, it turns out that the sequence of the mixture estimates obtained via the SEM is closer to the EM estimates than to the CEM estimates.

The SEM algorithm does not converge pointwise: the process generated by the SEM is a Markov chain whose stationary distribution is concentrated around the ML parameter estimator. A natural estimate $\boldsymbol{\phi}_h$ from a SEM sequence, $h = 1, ...$, is the mean of the iterates values where the first burn-in iterates have been discarded. An alternative estimate is to consider the parameter value leading to the highest likelihood in a SEM sequence.

The theoretical convergence properties of SEM algorithms are difficult to assess since it involves the study of the ergodicity of the Markov chain and the existence of the corresponding stationary distribution. Under regularity conditions, Diebolt and Celeux (1993) proved weak convergence to a local maximum. However, computational studies showed that the SEM algorithm is even better than EM algorithm for several cases, for example censored data (Chauveau, 1995), mixture models (Celeux, Chauveau and Diebolt, 1996). A drawback of this procedure is that it requires thousands of simulations and the computational cost could be, again, very high.

2.6 The choice of K

When discussing the process of parameters estimation, we assumed that the number K of mixture components is fixed; in practice, however, it is unknown and, thus, must be estimated through observed data. In this Section, we discuss methods for estimating K in a mixture model.

One of the big advantages of model-based clustering is that it provides a theoretical basis for estimating the number of clusters. This represents a great advantage with respect to the standard algorithms for clustering, where methods to determinate the number of clusters or the best clustering are questionable. Here, the problem is to perform a comparison among the members of a set of possible models.

It should be however noted that the choice of the number of components may not be a fundamental issue when mixture models are used to provide a semi-parametric estimate of an unknown distribution function: in this case over-estimation of the number of components is not a serious problem. When mixtures are used for distributional approximation purposes, Leroux

(1992) showed that, under very mild conditions, Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) along with other penalized likelihood criterion do not asymptotically underestimate the "true" number of components. Technically speaking, they are incorrectly applied because in this context the regularity conditions for the simple penalty functions based on the number of parameters do not hold.

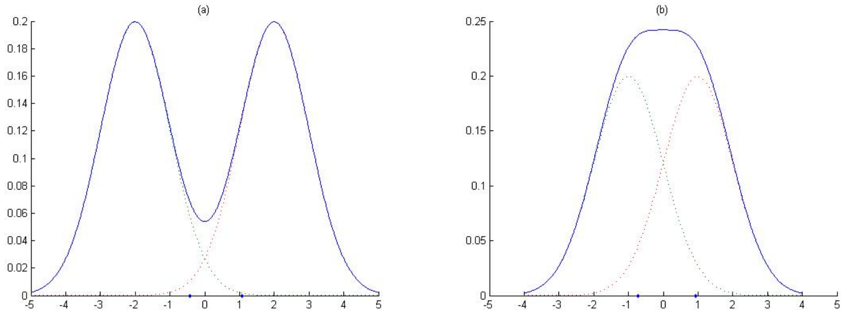
Our goal is to use the mixture model to provide model-based clustering and in this perspective the choice of the number of components must be carefully and critically addressed. The problem of over-fitting arises from the nesting of the mixture models, so that a distribution which can be well approximated by a mixture of K component densities can also be well approximated by a number $K_0 \geq K$ of component densities, in the sense that the two mixture distributions can be empirically indistinguishable. Thus, we can only get a lower bound for the number of components; in this respect the right question to ask is not "how many components are there in the mixture model?" but "what is the smallest number of the components in the mixture needed to make the model compatible with the observed data?". Especially, for model-based clustering, the answer to the latter would guarantee a reasonable explanation of the data generating process without being wasteful.

In the next Sections we will discuss different approaches to estimate the number of components in a mixture model. Obviously we will not try to be exhaustive, but give space to the more interesting approaches, while the interested reader can refer to the specific references.

2.6.1 Exploring the number of Modes

Estimating the number of components in a mixture distribution by analyzing the number of modes is one of the oldest methods, mainly based on intuition. Titterington et al. (1985) described some inferential procedures for assessing the number of modes. However, the obvious drawback of this method is that if the component densities are not sufficiently far apart the mixture distribution would still be unimodal and estimating the number of components by the number of modes would fail. Note however, that the practical interest could lie in finding components that correspond to separate modes, so that true separation occurs. Then the problem is: "do the components model departure from the parametric kernel density represent well distinct groups?"

Figure 2.2: Mixture of Gaussians (a) means 4 s.d. apart, (b) means 2 s.d. apart



We illustrate the distinction between modes and components through a simple example considering a mixture of two univariate Gaussian densities. Figure 2.2(a) is the mixture of two Gaussians with equal weights, with means being 4 standard deviation apart while Figure 2.2(b) shows the same mixture with means 2 standard deviations apart. In both figures dotted lines represent the component densities and the solid line is the the mixture density. Though Figure 2.2(a) produces a bimodal distribution, Figure 2.2(b) is still unimodal.

The example illustrates that we cannot always infer about the number of components using the number of modes. However number of modes $\leq K$ in univariate contexts; visually inspecting the modes of a multivariate distribution is moreover more difficult.

A method for assessing the separation between mixture components is based on the rootgrams of the posterior probabilities for the components of the mixture model. A rootogram is the variant of a histogram where the heights of the bars encode the square roots of the bin counts, instead of the bin counts themselves. This makes low counts more visible. If the rootogram for a certain k-th component has a large peak at $P(Z = k|Y)$=1 and is essentially zero elsewhere, this indicates clear separation of the k-th component from all the other components. On the other side, if the rootogram for the k-th component has no peak at $P(Z = k|Y)$=1 this is due to the fact that the k-th component is overlapped with other components. Furthermore, if there is significant mass away from $P(Z = k|Y)$=1 in the rootograms for several components, these components are not well separated.

2.6.2 Likelihood-based approaches

Likelihood based approaches are probably the most extensively used methods for testing statistical hypotheses. As noted before, model selection can be framed within a hypothesis testing problem. An obvious way of approaching the problem of testing the smallest value of the number of components in a mixture model is to use the LRT. We consider testing the null hypothesis H_0: $K = K_0$ versus H_1: $K = K_1$ for some $K_1 > K_0$, for example $K_1 = K_0 + 1$. Let $\hat{\boldsymbol{\phi}}_{K_i}$ be the MLE of $\boldsymbol{\phi}$ under H_i ($i = 0, 1$), and let $\lambda = \frac{L(\hat{\boldsymbol{\phi}}_{K_0})}{L(\hat{\boldsymbol{\phi}}_{K_1})}$ be the corresponding likelihood ratio. In the context of mixture models, it is well-known, that regularity conditions ($\Theta_0 \cap \Theta_1 = \emptyset$ and $\Theta = \Theta_0 \cup \Theta_1$) fail to hold for the likelihood ratio test statistic (LRT) $-2\log\lambda$ to have its asymptotic chi-squared null distribution with degrees of freedom equal to the difference between the number of parameters under the null and alternative hypotheses (Feng and McCulloch, 1996). In fact, in this context, the null hypothesis H_0 can be achieved from the alternative hypothesis H_1 or more formally the null hypothesis lie on the boundary of the alternative one. To explain this, let us consider the null hypothesis of one Gaussian component:

$$H_0 : f(y) = \varphi(y; \mu, \sigma^2)$$

versus the alternative hypothesis of two Gaussian components:

$$H_1 : f(y) = \pi\varphi(y; \mu_1, {\sigma_1}^2) + (1-\pi)\varphi(y; \mu_2, {\sigma_2}^2).$$

The model under H_0 can be obtained in two ways: with $\pi = 0$ or, respectively $\pi = 1$, $\mu_1 = \mu_2$ or with $\sigma_1 = \sigma_2$ and any π. Thus, the null hypothesis is nested within the alternative, i.e. $\Theta_0 \cap \Theta_1 \neq \emptyset$. In other words, Θ_0 is on the boundary of Θ_1.

McLachlan (1987) proposed a bootstrap approach, where the null distribution of $-2\log\lambda$ is estimated by fitting the mixture model to B samples drawn under the null hypothesis of K_0 components. That is, the bootstrap samples are generated from a mixture model with the vector $\boldsymbol{\phi}$ of unknown parameters replaced by its maximum likelihood estimate (MLE) under H_0, $\hat{\boldsymbol{\phi}}_{K_0}$. The value of $-2\log\lambda$ is computed for each bootstrap sample after fitting mixture models for $K = K_0$ and K_1 in turn to it. The process is repeated independently a number of times B, and the replicated values of $-2\log\lambda$ from bootstrap samples provide an assessment of the bootstrap null distribution of the test statistic. This approach can be used to compare the

LRT value on the original sample with a specific quantile of the bootstrap distribution, corresponding to a given level α. The value of the i-th order statistic of the B bootstrap replications can be used to estimate the quantile of order $i/(B+1)$, and the P-value can be evaluated referring to the distribution of the test statistic. McLachlan (1987) observed that the number of the bootstrap samples, B, must be very large if an accurate estimate of the P-value is pursued.

In details, if the decision to be taken concerns only the rejection of the null hypothesis at a significance level α, Aitkin, Anderson e Hinde (1981) underline that the bootstrap replications can be used to assess a test of approximate dimension α, likewise to the Monte Carlo procedure of Hope (1968). The test that rejects H_0 if the observed value of the statistic $-2\log\lambda$ is greater than the i-th ordered value in the B bootstrap replications has approximately dimension equal to:

$$\alpha = 1 - i/(B+1).$$

The results in McLachlan (1987) and in McLachlan and Peel (1997) show the performance of the bootstrap method in the case of limited dimension (few data and two or three clusters). The simulation studies of McLachlan and Peel (1997) illustrate that there is a tendency of the bootstrap approach to underestimate the upper percentiles of the null distribution of $-2\log\lambda$, and hence a tendency towards the null hypothesis of K_0 components (conservative test).

2.6.3 Bayesian and penalized likelihood approaches

A further approach to estimate the number of components is to use formal Bayesian and penalized likelihood methods (as the AIC and the BIC criteria). These methods are simple to be implemented since they are based on a penalization of the log-likelihood function through a simple additive factor. However, there are some theoretic limitations to apply those standard methods in the case of mixture models (Titterington et al., 1985).

The purely Bayesian approach considers the number of components K as a random variable with a known parametric distribution to obtain a posterior distribution for K conditional on the observed data. Often, this posterior distribution can not be calculated in closed form and must be approximated e.g. through MCMC techniques (Lavine and West, 1992; Diebolt and Robert, 1994). However, the two approaches which will be described in this Section can be viewed in a unique perspective since each penalized likelihood method can derive from a different approximation of the Bayesian solution (Chickering and Heckeman, 1997).

We consider the approach based on the Bayes factors and the posterior probabilities (Kass and Raftery, 1995). The idea of this approach is to consider K different models, M_1,..., M_K, with prior probabilities $p(M_k)$. By the Bayes theorem, the posterior probability of model M_k conditional on the observed data D is proportional to the product between the probability of the data conditional on the model M_k and the prior probability of the model:

$$p(M_k|D) \propto p(D|M_k)p(M_k).$$

When parameters are unknown, from the total probability principle, $p(D|M_k)$ is obtained integrating over the parameters, for example

$$p(D|M_k) = \int p(D|\boldsymbol{\theta}_k, M_k)p(\boldsymbol{\theta}_k|M_k)d\boldsymbol{\theta}_k,$$

where $p(\boldsymbol{\theta}_k|M_k)$ is the prior distribution of $\boldsymbol{\theta}_k$, the parameter corresponding to model M_k. The quantity $p(D|M_k)$ is the so-called *integrated likelihood* of model M_k.

Obviously, a natural Bayesian approach to select a model is to choose the model with the largest posterior probability, and if the prior probabilities of the model are the same, choosing the model with the largest integrated likelihood. To compare two models, M_1 and M_2, the Bayes factor is defined as the ratio of the corresponding integrated likelihoods:

$$B_{1,2} = \frac{p(D|M_1)}{p(D|M_2)}.$$

In others words, the Bayes factor represents the posterior odds that the data are distributed according to model M_1 rather than according to model M_2 when both models have the same prior probabilities. If $B_{1,2} > 1$, model M_1 is to be preferred to model M_2. Conventionally, for values of $B_{1,2}$ greater than 100 the choice is strongly motivated; obviously, the method can be generalized to more elements.

The main difficulty in using the Bayes factor is the calculation of the integrated likelihood. In 1973, Akaike proposed an approximation of the integrated likelihood, which is often referred to the Akaike Information Criterion (AIC). Bozdogan and Sclove (1984) and Sclove (1987) developed AIC in the context of selection of the number of components of a mixture models as follows:

$$-2 \log p(D|\hat{\theta}_k; M_k) + 2\nu_k = AIC_k$$

where ν_k is number of the parameters in model M_k and $\hat{\theta}_k$ is the ML estimate of θ_k. However, Soromenho (1993) and Celeux and Soromenho (1996) proved that AIC tends to overestimate the correct number of components.

Schwarz (1978) proposed a different approximation of the integrated likelihood, called the Bayesian Information Criterion (BIC), which is given by:

$$-2\log p(D|\hat{\theta}_k; M_k) + \nu_k \log(n) = BIC_k.$$

The first term of such equation, which represents the maximized likelihood of the mixture, is penalized to restrain the tendency towards unnecessary parameters proliferation. A minimum value of BIC shows a strong evidence for the corresponding model; finally, BIC can be used to compare models with different parameterizations of the covariance matrices and different number of clusters. Usually, differences of BIC greater than 10 are considered as strong evidence to support a specific model (Kass and Raftery, 1995).

Another criterion is the so-called AWE, *approximate weight of evidence* (Banfield and Raftery, 1993). It has been derived by an approximation of the integrated likelihood based on the classification likelihood. However, empirical results seem to encourage the use of BIC (Fraley and Raftery, 1998).

2.6.4 The use of information criteria

The problem of model selection can be approached using some information criteria. In fact, both AIC and BIC have some links with the Kullback-Leibler divergence (1951) of a unknown density function $f(\mathbf{y})$ from its estimate $f(\mathbf{y}; \hat{\boldsymbol{\phi}})$ given by:

$$I[(f(\mathbf{y}); f(\mathbf{y}; \hat{\boldsymbol{\phi}})] = \int f(\mathbf{y}) \log f(\mathbf{y}) d\mathbf{y} - \int f(\mathbf{y}) \log f(\mathbf{y}; \hat{\boldsymbol{\phi}}) d\mathbf{y} \tag{2.13}$$

and with the bias-corrected log-likelihood:

$$\log L(\hat{\boldsymbol{\phi}}) - b(F) \tag{2.14}$$

where $b(F)$ is the bias of the estimator of the second term in the right hand side of (2.13), the only important term since the first one does not depend on the model. The purpose is to select the model (i.e., the number of components in this context) which maximizes expression (2.14) and, therefore, minimizes the divergence (2.13). In literature, information criteria are expressed in the following form:

$$-\log L(\hat{\boldsymbol{\phi}}) + 2C \tag{2.15}$$

where the first term measures lack of fit while the second represents a penalty term which measures model complexity. Therefore, the aim is to choose the model which minimizes the criterion (2.15).

An alternative method based on information criteria which has a similar computational complexity to the bootstrap method of McLachlan (1987) is the *cross-validation* method which has been shown by Smyth (2000) to work well in practical examples.

To explain, let $\mathbf{y}$ be a random vector with density function $f(\mathbf{y})$ and let $D^{train} = \{\mathbf{y}_1, ..., \mathbf{y}_n\}$ be a random sample from the analyzed sample. A set of finite mixture models with $k = 1, ..., K$ density components are fitted to $D^{train} = \{\mathbf{y}_1, ..., \mathbf{y}_n\}$. Thus, we have an indexed set of estimated models, $f(\mathbf{y}; \hat{\boldsymbol{\phi}}^{(k)})$, $1 \leq k \leq K$, where each $f(\mathbf{y}; \hat{\boldsymbol{\phi}}^{(k)})$ is estimated on the same data set D^{train}. Let

$$l_k^{train} = l(\hat{\boldsymbol{\phi}}^{(k)} | D^{train})$$

be the log-likelihood function of the model with $k = 1, ..., K$ components, where $\hat{\boldsymbol{\phi}}^{(k)}$ are the ML estimates obtained from the data D^{train}. l_k^{train} is a non decreasing function of k since model flexibility increases with increasing number of density components, i.e. with better fit to the data. Therefore l_k^{train} can not directly provide any indication for the number of components in the mixture model.

Let us imagine that one has a large data set D^{test} which was not used in fitting any of the adopted models; in general, $D^{test} \cup D^{trai} = D$ where D is the observed sample.

Let

$$l_k^{test} = l(\hat{\boldsymbol{\phi}}^{(k)}|D^{test})$$

be the log-likelihood function evaluated on data D^{test} using the parameter set estimates obtained from D^{train}. We can interpret this test log-likelihood function as a function of k, keeping all other parameters as well as D^{train} fixed. Intuitively, this test likelihood l_k^{test} should be a useful test statistic for comparing mixture models with different numbers of components. This test log-likelihood is also known as the log predictive score, Good (1952).

For convenience of notation, let $f_k(\mathbf{y})$ denote the k-components model with parameter estimates $\hat{\boldsymbol{\phi}}^{(k)}$, and let

$$i_k = -\frac{l_k^{test}}{n_{test}} = -\frac{1}{n_{test}} l(\hat{\boldsymbol{\phi}}^{(k)}|D^{test})$$

be the negative test log-likelihood per unit sample. The expected value of i_k is equal to

$$E(i_k) = -\frac{1}{n_{test}} E[l(\hat{\boldsymbol{\phi}}^{(k)}|D^{test})] = -\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} E[\log f_k(\mathbf{y}_i)]$$

$$-E[\log f_k(\mathbf{y}_i)] = \int f(\mathbf{y}) \log \frac{1}{f_k(\mathbf{y})} d\mathbf{y} = \int f(\mathbf{y}) \log \frac{f(\mathbf{y})}{f_k(\mathbf{y})} d\mathbf{y} + \int f(\mathbf{y}) \log \frac{1}{f(\mathbf{y})} d\mathbf{y} \tag{2.16}$$

where n_{test} is the dimension of D^{test}. This expected value is the sum of the Kullback-Leibler divergence between $f(\mathbf{y})$ and $f_k(\mathbf{y})$ (the first term on the right) and a constant which is independent of k and represents the entropy of the true density function $f(\mathbf{y})$ (the second term on the right above). Therefore, the test log-likelihood function l_k^{test}, when appropriately scaled, is an unbiased estimator (unless a constant) of the Kullback-Leibler divergence (KL). The KL divergence in turn defines how far the model $f_k(\mathbf{y})$ is from the true f and is strictly positive unless $f(\mathbf{y}) = f_k(\mathbf{y})$. Thus, the test log-likelihood function is au unbiased estimator of the KL divergence between the true density and the model under consideration, and this motivates its use as a model selection criterion. Often, we have not a large independent test data to be used as D^{test}; thus, a practical alternative is to use *cross-validation* to select the model. The data are repeatedly partitioned into two sets, one is used to estimate model parameters and the other is used to evaluate the statistic of interest. Let M be the number of partitions. For the i-th partition let S_i be the data subset used for evaluation of the log-likelihood function and $D\S_i$ be the data used for building the model. Therefore, the cross-validated estimate of the test log-likelihood function for k-th model is defined as:

$$l_k^{CV} = \frac{1}{M} \sum_{i=1}^{M} l(\hat{\boldsymbol{\phi}}^{(k)}(D\S_i)|S_i) \tag{2.17}$$

where $\hat{\boldsymbol{\phi}}^{(k)}(D\S_i)$ denotes the parameters for the k-th model estimated from the i-th training subset, and the term within the sum of the equation (2.17) is the log-likelihood function evaluated on the data in S_i using the parameters estimated from the data D/S_i.

In general, consider the case when the model family under consideration includes the true data generating distribution $f(\mathbf{y})$; let this particular model have k_{true} density components. Both the Bayesian and cross-validation methods will tend to converge to k_{true} as the sample size increases. For cases where truth is not within the model family, it is clear from the KL equations above that the cross-validation methodology will directly seek that particular model within the chosen model family which is closest to truth.

There are different techniques of cross-validation that differ in how the partition are chosen. For example, v*-fold* cross-validation uses v disjoint test partitions $\{S_1, ..., S_v\}$ each of size n/v. Well known examples are: $v = n$ (*leave-one-out*) and $v = 10$ (ten-fold CV, Breiman et al., 1984).

Chapter 3

Simultaneous clustering and factorial reduction

This Chapter deals with approaches to simultaneous clustering and factorial reduction. It represents an attempt to solve the task of clustering few units on a very large number of the variables; this is a non standard problem in statistic analyses, which is, however, usual in some fields of research, where high-dimensional data are considered (for example, marketing, customer satisfaction, psychology, document classification and gene expression data analyses). In particular, if a mixture model is adopted to cluster few units on a large number of variables, conceptual and computational limitations make the use of standard algorithms for maximum likelihood estimation rather cumbersome. In standard contexts, observations are usually considered as n independent realizations of a J-dimensional random variable; here, the sample size is assumed to be greater than the number of variables in order to avoid near-singular estimates of the (component-specific) covariance matrices. Therefore, if the aim is to cluster few observations characterized by a very large number of variables, we may face some problems with parameter estimation; this leads to the need of some factorial reduction prior to a clustering algorithm.

Principal Component Analysis (PCA) has often been applied to reduce the number of analyzed variables, to apply a clustering algorithm for grouping units into homogeneous groups on the basis of principal components. The hope for using PCA prior to cluster analysis is that PCs may "extract" the essential information about the cluster structure in the analyzed data set. Since PCs are uncorrelated and ordered, the first few PCs, which contain most data variability (information), are usually used in cluster analysis (see e.g. Jolliffe et al., 1980). However, De Sarbo et al. (1990) and De Soete and Carroll (1994) among others warn against this procedure, which is referred to as "*tandem analysis*" (Arabie and Hubert, 1994); it provides PCs and an optimal classification, minimizing two target functions that may work in opposite direction. That is because PCA may identify PCs that do not contribute much to perceive the clustering structure in the analyzed data but, on the contrary, may obscure or mask it.

To overcome this problem, several authors have proposed techniques for simultaneous clustering and factorial reduction of the analyzed data (see e.g. De Soete and Carroll, 1994; Tipping and Bishop, 1997; Vichi and Kiers, 2001).

In the next Sections, we examine and compare the models proposed by Ghahramani and Hinton (1996) (successively extended by McLachlan et al., 2000b) and Rocci and Vichi (2002) for the purposes of analyzing high dimensional data in a lower dimensional space to explore group structures. The comparison is discussed with respect to the adopted reparameterization of the factorial representation. A comparison on gene expression data can be found in Martella (2006).

3.1 Ghahramani and Hinton (1996)

Ghahramani and Hinton (1996) propose a model that simultaneously performs clustering and local dimensional reduction within each cluster. Using a factorial approach, the J-dimensional data vector $\mathbf{y}_i$ is modeled as

$$\mathbf{y}_i = \boldsymbol{\mu} + \mathbf{B}\mathbf{u}_i + \mathbf{e}_i, \tag{3.1}$$

where $\boldsymbol{\mu}$ is a (overall) mean vector, $\mathbf{B}$ is a $J \times Q$ matrix of factor loadings, $\mathbf{u}_i$ is a Q-dimensional ($Q<J$) vector of latent variables (known as factors), which are assumed to be i.i.d. draws from a $N(\mathbf{0}, \mathbf{I}_Q)$, where $\mathbf{I}_Q$ denotes the $Q \times Q$ identity matrix. Furthermore, $\mathbf{e}_i$ are i.i.d. random variable with Gaussian distribution $N(\mathbf{0}, \mathbf{D})$, where $\mathbf{D} = diag(\sigma_1^2, ..., \sigma_J^2)$, that are assumed to be independent of $\mathbf{u}_i$.

According to this model, conditional on $\mathbf{u}_i$, the distribution of $\mathbf{y}_i$ is $N(\boldsymbol{\mu} + \mathbf{B}\mathbf{u}_i, \mathbf{D})$, while unconditionally, the distribution of $\mathbf{y}_i$ is given by

$$\mathbf{y}_i \sim \ N(\boldsymbol{\mu}, \mathbf{B}\mathbf{B}' + \mathbf{D}). \tag{3.2}$$

If Q is chosen sufficiently smaller than J, the representation $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}' + \mathbf{D}$ imposes some constraints on the covariance matrix and thus reduces the number of free parameters to be estimated. However, in the case of $Q > 1$, $\mathbf{B}$ is not identifiable univocally, since $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}' + \mathbf{D}$ is still satisfied if $\mathbf{B}$ is replaced by $\mathbf{B}\mathbf{C}$, where $\mathbf{C}$ is any orthogonal matrix of order Q. One (arbitrary) way of uniquely specifying $\mathbf{B}$ is to choose the orthogonal matrix $\mathbf{C}$ so that $\mathbf{B}'\mathbf{D}^{-1}\mathbf{B}$ is diagonal (with its diagonal elements arranged in decreasing order). Assuming that the eigenvalues of $\mathbf{B}\mathbf{B}'$ are positive and distinct, the condition that $\mathbf{B}'\mathbf{D}^{-1}\mathbf{B}$ is diagonal imposes $Q(Q-1)/2$ constraints. Hence, the number of free parameters is $JQ + J - Q(Q-1)/2$.

The goal of factor analysis is to find those $\mathbf{B}$ and $\mathbf{D}$ that best fit the covariance structure of $\mathbf{y}_i$. Ghahramani and Hinton (1996) considered a mixture of

K factor models (3.1), called mixture of factor analyzers ($k = 1, ..., K$); each factor analyzer depends on a set of Q latent factors through a component-specific factor loading matrix $\mathbf{B}_k$ and show different mean vectors $\boldsymbol{\mu}_k$. Thus, each $\mathbf{y}_i$ is a mixture of K Gaussian densities in proportions $\pi_1, ..., \pi_K$; that is

$$f(\mathbf{y}_i; \boldsymbol{\phi}) = \sum_{k=1}^{K} \pi_k \varphi(\mathbf{y}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.3}$$

where $\boldsymbol{\Sigma}_k = \mathbf{B}_k \mathbf{B}'_k + \mathbf{D}_k$, ($k = 1, ..., K$).

Thus, the set of model parameters in given by

$$\boldsymbol{\phi} = \{(\boldsymbol{\mu}_k, \mathbf{B}_k)_{k=1,...,K}, \boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_{K-1}, \mathbf{D}_k\}.$$

This model has been further developed by McLachlan et al. (2000b). They fitted the mixture of factor analyzers model by using the AECM algorithm (see Section 2.5.1). To apply the AECM algorithm to the mixture of factor analyzers, McLachlan et al. (2000b) partition the vector of unknown parameters $\boldsymbol{\phi}$ as ($\boldsymbol{\phi}_1$; $\boldsymbol{\phi}_2$), where $\boldsymbol{\phi}_1$ contains the mixing proportions π_k ($k = 1, ..., K$) and the elements of the component means $\boldsymbol{\mu}_k$ ($k = 1, ..., K$). The subvector $\boldsymbol{\phi}_2$ contains the elements of $\mathbf{B}_k$ ($k = 1, ..., K$) and $\mathbf{D}_k$. One iteration of the AECM algorithm consists of two cycles: one E-step (in the same form of the standard one for Gaussian mixture models) and one CM-step for each cycle. The two CM-steps correspond to the partition of $\boldsymbol{\phi}$ into the two sub-vectors $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. In the first cycle, the CM-step leads to π_k and $\boldsymbol{\mu}_k$ being updated as in (2.7) and (2.8). In the second cycle, the CM-step leads to updated estimates of $\mathbf{B}_k$ ($k = 1, ..., K$) as

$$\mathbf{B}_k = \mathbf{A}_k \gamma_k (\gamma'_k \mathbf{A}_k \gamma_k + \omega_k)^{-1},$$

where

$$\mathbf{A}_k = \frac{\sum_{k=1}^{K} w_{ik}(\mathbf{y}_i - \boldsymbol{\mu}_k)(\mathbf{y}_i - \boldsymbol{\mu}_k)'}{\sum_{k=1}^{K} w_{ik}},$$

$$\gamma_k = (\mathbf{B}_k\mathbf{B}_k' + \mathbf{D}_k)^{-1}\mathbf{B}_k'$$

and

$$\omega_k = \mathbf{I}_Q - \gamma_k'\mathbf{B}_k$$

$k = 1, ..., K$. The updated estimates $\mathbf{D}_k$ are given by:

$$\mathbf{D}_k = diag\{\mathbf{A}_k - \mathbf{A}_k\gamma_k\mathbf{B}_k'\}.$$

It can be proved that some of the estimates of $\mathbf{D}_k$ will be close to zero if effectively not more than Q observations are assigned to the k-th component of the mixture (in terms of the fitted posterior probabilities of component membership). This will lead to spikes or near singularities in the likelihood function. One way to overcome this problem is to impose the condition of a common value $\mathbf{D}$ for the $\mathbf{D}_k$,

$$\mathbf{D}_k = \mathbf{D} \qquad (k = 1, ...K).$$

It is worth noticing that the mixture of factor analyzers is, essentially, a mixture of reduced Gaussian distributions, where a component specific factor model fits a Gaussian distribution to a portion of data, weighted by posterior probabilities, w_{ik} ($k = 1, ..., K$, $i = 1, ..., n$). Since the covariance matrix for each component is specified through the lower dimensional factor loading matrices, the model has $[JQ - (Q^2 - Q)/2]K + J$ rather than $KJ(J + 1)/2$ parameters.

If Q is sufficiently smaller than J, $\mathbf{\Sigma}_k = \mathbf{B}_k\mathbf{B}_k' + \mathbf{D}$ with $\mathbf{C}_k$, a orthogonal matrix of order Q, such that $\mathbf{B}_k'\mathbf{D}^{-1}\mathbf{B}_k$ is diagonal imposes stronger restrictions on covariance matrices reducing the number of parameters to be estimated.

McLachlan et al. (2006) suggest the use of K factor models with mixtures of t distributions in attempt to make the model less sensitive to outliers.

3.2 Rocci and Vichi (2002)

Rocci and Vichi (2002) proposed a clustering model which has insightful links with the mixture of factor analyzers proposed by Ghahramani and Hinton (1996). More precisely, they propose a two-way model for simultaneous factorial reduction and clustering assuming that the observed data come from a finite mixture of multivariate Gaussian distributions. Constraining the mean vectors of each component density to lie onto a subspace, model parameters are estimated through maximum likelihood using an Expectation Conditional Maximization (ECM) algorithm.

They consider the mixture model (2.3) and represent the mean vector $\boldsymbol{\mu}_k$=$[\mu_{k1}, ..., \mu_{kJ}]$ of each component density, as a function of $Q < J$ latent variables according to the following bilinear model:

$$\boldsymbol{\mu}_k = \mathbf{B}\mathbf{u}_k, \tag{3.4}$$

where $\mathbf{B}$ is a factor-loading matrix $[J \times Q]$, while $\mathbf{u}_k = [u_{k1}, ..., u_{kQ}]$ represents component-specific factor scores.

Let $\mathbf{y}_1, ..., \mathbf{y}_n$ be n i.i.d. observations, the maximized log-likelihood of the mixture model (2.3) can be written as (Hathaway, 1986):

$$\begin{aligned} l(\hat{\boldsymbol{\phi}}) = \sum_{i=1}^{n} \log[\sum_{k=1}^{K} \hat{\pi}_k f(\mathbf{y}_i; \hat{\boldsymbol{\theta}}_k)] = \\ = \sum\nolimits_{ik} w_{ik} \log[\hat{\pi}_k f(\mathbf{y}_i; \hat{\boldsymbol{\theta}}_k)] - \sum\nolimits_{ik} w_{ik} \log(w_{ik}), \end{aligned} \tag{3.5}$$

where $\hat{\boldsymbol{\phi}} = (\hat{\pi}_k, \hat{\boldsymbol{\theta}}_k)$ denotes the maximum likelihood estimate of the parameter vector, while w_{ik} represents the posterior probability that the i-th unit belongs to the k-th component of the mixture subject to the constraint $\sum_{k=1}^{K} w_{ik} = 1$ $(i = 1, ..., n)$. Rocci and Vichi (2002) propose to estimate the parameters model by using a ECM type algorithm (see Section 2.5.1). The algorithm can be formulated by maximizing the log-likelihood function with respect to only a subset of model parameters conditionally upon the others.

In this algorithm the estimates of w_{ik}, π_k and $\boldsymbol{\Sigma}_k$ have the same form of a standard mixture model; the estimates of $\mathbf{B}$ and $\mathbf{u}_k$ are respectively given by:

$$vec(\mathbf{B}) = \{\sum_k n\pi_k[(\mathbf{u}_k\mathbf{u}_k') \otimes \boldsymbol{\Sigma}_k^{-1}]\}^{-1} vec(\sum_{ik} w_{ik}\boldsymbol{\Sigma}_k^{-1}\mathbf{y}_i\mathbf{u}_k').$$

and

$$\mathbf{u}_k = (\mathbf{B}_k'\boldsymbol{\Sigma}_k^{-1}\mathbf{B}_k)^{-1}\mathbf{B}_k'\boldsymbol{\Sigma}_k^{-1}\frac{\sum_i w_{ik}\mathbf{y}_i}{n\pi_k}.$$

It is worth noticing that the proposed model can be considered a fuzzy version of the model suggested by De Soete and Carroll (1994), where each cluster has (in addition) a specific Mahalanobis metric.

3.3 Conclusions

We have examined two models allowing for simultaneous clustering and factorial reduction of the analyzed data proposed by Ghahramani and Hinton (1996) and Rocci and Vichi (2002), respectively.

The main difference is that the latter assumes that the mean vectors lie in a common reduced subspace in order to identify those latent factors that best explain the between groups variability (i.e. the variables with the greatest discriminant power). The model proposed by Rocci and Vichi (2002) can be used when the number of units is lower than the number of variables (i.e. $n \leq J$) only imposing some restriction on the covariance matrices. On the other hand, the former approach assumes unconstrained component-specific mean vector and controls the number of parameters by modelling the component-specific covariance matrices, $\mathbf{\Sigma}_k = \mathbf{B}_k\mathbf{B}_k' + \mathbf{D}$ $(k = 1, ..., K)$, explaining the correlations between variables through the latent factors, $\mathbf{u}$. This provides an intermediate model between the independence model and the unrestricted model which is able to capture some interesting structures in the data, without fitting a full covariance matrix.

It is worth noticing that the number of parameters to be estimated in the model of Ghahramani and Hinton (1996) is

$$[JQ - (Q^2 - Q)/2]K + J + JK + K - 1$$

while those to be estimated in Rocci and Vichi (2002) model are

$$JQ + KQ - Q^2 + K - 1 + [KJ(J + 1)/2].$$

If we assume a common and spherical covariance matrix $\mathbf{\Sigma}$ in the model of Rocci and Vichi (2002) and a covariance matrix $\mathbf{\Sigma} = \mathbf{B}\mathbf{B}' + \mathbf{D}$ in the model of Ghahramani and Hinton (1996), the number of parameters of Rocci and Vichi (2002) $(JQ + KQ - Q^2 + K - 1 + 1)$ is considerably lower than the number of parameters of Ghahramani and Hinton (1996) $(JQ - (Q^2 - Q)/2 + J + JK + K - 1)$. In other words, if it is possible to assume a common and spherical covariance matrix for the density function of the analyzed data, the model of Rocci and Vichi (2002) has a lower computational complexity.

The interested readers are refereed to Martella (2006) for the analysis of a benchmark data set on gene expression data.

Chapter 4

Double clustering

We will discuss different approaches to double clustering, that is, methods that provide a simultaneous clustering of the rows and of the columns of a data matrix.

In general, double clustering methods can be useful in a broad range of applications where the aim is to identify *blocks* (or biclusters), i.e., sub-matrices of the observed data matrix, which satisfy some specific characteristics of homogeneity. The characteristics of homogeneity characterizing each block may vary in different approaches. Moreover, units and variables forming each block specify an unit cluster and a variable cluster.

Such methodologies are known under a broad range of names, including *direct clustering*, *biclustering*, *block clustering*, *bidimensional clustering*, *subspace clustering co-clustering*, *simultaneous clustering* and *blockmodeling.* However, it has to be noticed that these terms highlight different features of the clustering approaches. For example, if there are relationships between units and variables clusters, and clustering of one dimension is dependent on the clustering of the other, this form of clustering is often referred to as *two-way clustering* (see for example Getz et al., 2000; Tang et al., 2001;

Pollard and van der Laan, 2002 and Getz et al., 2003). The concept of *co-clustering* (or simultaneous clustering), instead, has been introduced by Dhillon (2001) in the context of document-keyword analysis and indicates a form of two-way clustering in which both dimensions are clustered simultaneously; Kluger et al. (2003) proposed a similar method for co-clustering gene expression data. Although they refer to their method as *spectral biclustering*, it could be differentiated from biclustering, as the clusters are dependent on the full expression profile of genes or samples, and the clustering leads to exhaustive, non-overlapping clusters. Finally, the term *biclustering* is often used (see Cheng and Church, 2000; Segal et al. 2001; MacKay and Miskin, 2001; Tanay et al., 2002; Lazzeroni and Owen, 2002; Ben-Dor et al., 2002; Segal, 2003; Ambler, 2003) to identify possibly overlapping sub-matrices of the data that exhibit interesting homogeneous blocks, leaving the remaining data unclustered. Thus biclustering may be viewed as an extension of context-specific one-way clustering. It combines the features of iterative two-way clustering and co-clustering, in that local dependencies can be discovered, but the analysis is based on the full expression matrix and units and variables are clustered simultaneously.

Throughout this dissertation, we use the term *double clustering* in order to refer to the methods in aimed at finding a set of homogeneous blocks in a matrix.

Double clustering approaches are often needed since methods based on simultaneous clustering and factorial reduction of a data matrix may fail in detecting relevant information in the data. In particular, in microarray analysis, a major problem consists in clustering patients or tissues (in general, experimental conditions) with similar behaviour with respect to genes expressions. However, applying clustering and factorial reduction techniques to experimental conditions leads to significant difficulties. In fact, many activation patterns are common to groups of genes only under specific experimental conditions. Therefore, double clustering of rows and columns allows to achieve the further goal of detecting groups of genes with similar functions characterizing a specific subset of experimental conditions.

Alternative solution could be to apply the *Variable Selection for Clustering* methods, which allow to specify clusters based on a subset of the variables (see e.g. Devaney and Ram, 1997; McCallum, Nigam, and Ungar, 2000; Brusco and Cradit; 2001). Friedman and Meulman (2004) use this type of approach in the context of clustering tissue samples, allowing for the situation where only a small proportion of the genes are useful in distinguishing a particular cluster. Their procedure, Clustering Objects on Subsets of Attributes (COSA) computes distances between tissue samples, giving the expression levels gene- and sample-specific weights. These distances are then passed to a distance-based clustering algorithm, such as hierarchical clustering, to cluster the tissue samples. This will identify clusters characterized by a common profile. The structure of the clustering will depend on the method used. Thus the form and structure of the clusters is conventional, but the importance of each gene in the discovery of a sample cluster can be quantified and relevant genes can be isolated. Since the variables are weighted, rather than selected or removed, there is no actual dimension reduction although it does allow emphasis on different variables for different clusters. A similar idea in terms of weighting variables but with a different function to be optimized is suggested by DeSarbo, Carroll, Clarck and Green (1984).

4.1 Background

Statistical literature provides many clustering techniques for the identification of “homogeneous” groups of units which are perceived as “similar” to one another within each group (see e.g. Gordon, 1999). They can be applied to obtain a clustering of variables as well. If the interest is to cluster both units and variables these methods can be applied both to units and to variables successively and independently (see Tryon, 1939). However, results depend on whether units or variables are classified first.

To overcome this problem, Fisher (1969) proposed to partition units and variables simultaneously rather than successively. The most important advantage of a double (rather than a sequential) clustering is that the former allows to highlight the eventual interaction or dependence between units and variables helping in their characterization by using an “overall” objective function that cannot be reduced to a simple combination of row and column objective functions. When applying double clustering, variables should be expressed in the same scale of measurement, so that entries are comparable among both rows and columns; if this is not the case, data need to be adequately rescaled.

Starting with the pioneering work of Hartigan (1972) and with some decision-theoretic work by Bock (1974), during the past three decades this class of methods has been widely developed by various authors in different fields such as marketing, customer satisfaction, social network, psychology, text mining, election and nutritional analyses. Recently, double clustering techniques underwent increasing interest due to the challenge of finding suitable methods of analysis for microarray gene expression data in the bioinformatics context.

Double clustering methods are very heterogeneous both in terms of mathematical structures and underlying models, and in terms of principles and tools used in the data analysis step. As a consequence, it is clear that the corresponding domain has never been easily accessible.

In the last ten years, two structured taxonomies of double clustering methods have been proposed by Van Mechelen, Bock and De Boeck (2004) and by Madeira and Oliveira (2004). Van Mechelen, Bock and De Boeck (2004) built their overview starting from a traditional statistical/data analytic perspective while Madeira and Oliveira (2004) discuss this topic on the bioinformatics side.

We will not attempt to give an exhaustive overview of double clustering methods; the interested reader can refer to detailed reviews in Van Mechelen, Bock and De Boeck (2004) and Madeira and Oliveira (2004). We rather prefer giving a schematic explanation of key principles underlying these taxonomies.

4.2 Van Mechelen, Bock and De Boeck (2004) taxonomy

Van Mechelen, Bock and De Boeck (2004) identify three cluster types: row, column and data clusters (so-called blocks) (see Figure 4.1). Each of these clusters may have different nature and be obtained through distinct approaches. Thus, their overview is based on two structuring principles:

- set-theoretical nature of row, column and data clusters;
- type of model structure or associated loss function.

As for the set-theoretical nature of row, column and data clusters, they distinguished among partitions, nested clusters and overlapping clusters (see

Figure 4.2). A *partition* consists of a certain number of non-empty, non-overlapping clusters that span the full observed set. *Nested clusters* include intersecting clusters (an important special case being hierarchical clustering). Finally, *overlapping clusterings* include intersecting, non-nested clusters. Obviously, the nature of row, column, and data clusters can be different (some examples as showed in the Figure 4.3).

As far as the level of modeling and optimization is concerned, they discerned three main levels:

1. *Procedural level*: clustering algorithms that are neither based on loss functions to be optimized or on mathematical model structures;

2. *Deterministic level*: clustering algorithms based on a deterministic model to be fitted to data in order to optimize some overall loss function;

3. *Stochastic level*: clustering algorithms based on a stochastic model (that is, implying distributional assumptions) to be fitted to data in order to optimize some global criterion.

Figure 4.1: Example of row clusters, column clusters, data clusters

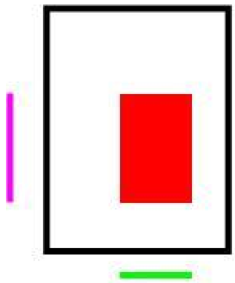
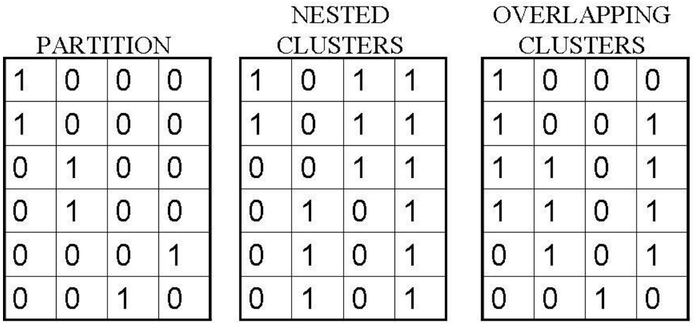


Figure 4.2: Hypothetical set-theoretical nature of clusters



PARTITION

1	0	0	0
1	0	0	0
0	1	0	0
0	1	0	0
0	0	0	1
0	0	1	0

NESTED CLUSTERS

1	0	1	1
1	0	1	1
0	0	1	1
0	1	0	1
0	1	0	1
0	1	0	1

OVERLAPPING CLUSTERS

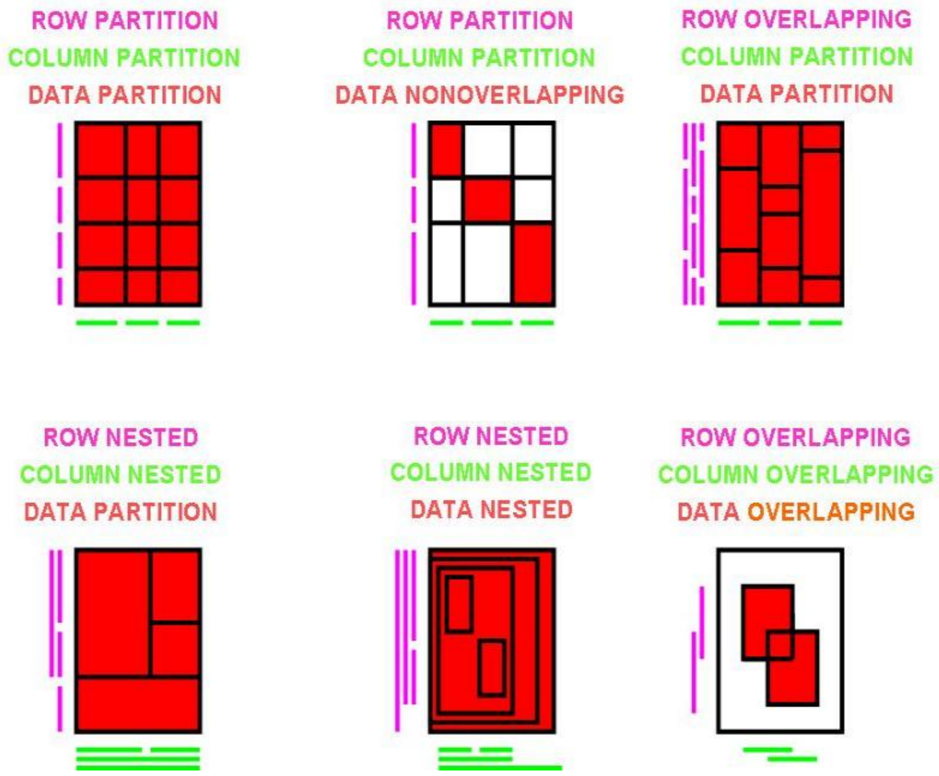
1	0	0	0
1	0	0	1
1	1	0	1
1	1	0	1
0	1	0	1
0	0	1	0

All methods that imply row/column partitions aim at approximating the $(n \times J)$ data matrix $\mathbf{X}$ by some $(n \times J)$ matrix $\hat{\mathbf{X}}$. The generic element $\hat{\mathbf{x}}_{ij}$ of $\hat{\mathbf{X}}$ is assumed to be constant within each data block. Thus, the values of the approximating matrix $\hat{\mathbf{X}}$ can be summarized by $(K \times Q)$ matrix $\bar{\mathbf{X}}$ of block constants, with

$$\hat{\mathbf{X}} = \mathbf{U}\bar{\mathbf{X}}\mathbf{V}', \tag{4.1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are membership matrices for row and column partitions. Notice that all methods based on procedural perspectives do not involve a global loss or objective function, but only permutations of rows and columns in order to find an optimal permutation (Arabie et al., 1988 propose e.g. the *bond energy algorithm*). Deterministic methods, instead, are based on a specific loss function and on different specifications of $\bar{\mathbf{X}}$; for example Govaert (1980) specified a binary matrix while Vichi (2001) specifies an arbitrary, real-valued matrix $\bar{\mathbf{X}}$. Finally, stochastic methods assume a parametric specification for both dimensions; for example Govaert and Nadif (2003) propose a stochastic extension of model (4.1) with arbitrary matrix $\bar{\mathbf{X}}$, latent (independent) partitions $\mathbf{U}$ and $\mathbf{V}$ where observations are drawn from i.i.d. blocks

Figure 4.3: Schematic representation of hypothetical examples of clustering that imply different row, column, data clusters



with known parametric distribution $P(\theta_{kq})$.

Methods that imply nested row/column clusterings can be distinguished into three subgroups in terms of the "shapes" of data clusters. In the first subgroup, data clusters take the form of partitions while in the second and third subgroups, they take the form of nested clusters. The difference between the second and the third subgroup pertains to additional restrictions with regards to nesting structure.

In particular, Hartigan (1975) specified two methods that do not involve an approximating matrix $\hat{\mathbf{X}}$ for the data matrix $\mathbf{X}$, but rather look for data clusters such that, for each data cluster, the column-variance (and optionally also the row-one) is not greater than a user-prespecified threshold. More specifically, the first method (one-way clustering) only aims at finding low within-column variances, while the second method (two-way joining) looks for both low column and row variances.

Finally, all methods that imply a restricted nested data clustering are based on a hierarchical structure made up of the disjoint union of row and column dimensions.

Methods implying overlapping row/column and data clusters are classified as procedural and deterministic methods. The main category is the deterministic one where a matrix $\hat{\mathbf{X}}$ is used to approximate the data matrix $\mathbf{X}$ as well as possible (using a L_1 or L_2 norm). The entries in $\hat{\mathbf{X}}$ are defined as cluster specific constants which can be derived either in a *simple additive* or a *boolean way.*

In particular DeSarbo (1982) introduced GENNCLUS (GENeral Nonhierarchical CLUStering); having the following additive form:

$$\mathbf{X} = \mathbf{U}\bar{\mathbf{X}}\mathbf{V}' + \mathbf{C}, \tag{4.2}$$

where $\mathbf{U}$ and $\mathbf{V}$ denote overlapping row and column clusters, $\mathbf{C}$ is a constant value matrix and $\bar{\mathbf{X}}$ is constrained to be square and symmetric. Finally, the Boolean approaches can be described as:

$$\bar{\mathbf{X}} = \mathbf{U} \otimes \mathbf{V}', \tag{4.3}$$

where $\otimes$ indicates a Boolean matrix product. Mickey et al. (1983) developed an iterative algorithm to fit (4.3) to a given binary matrix using a least squares loss function.

A schematic overview of this taxonomy is given in Figure 4.4, where the first column indicates one example from the set of suggested approaches.

Authors	Set-theoretical nature row-column clustering	Set-theoretical nature data clustering	Level of modeling and optimization
Arabie et al. (1988)	partition	partition	procedural
Vichi (2001)	partition	partition	deterministic
Govaert and Nadif (2003)	partition	partition	stochastic
Hartigan (1975) one-way spitting	nested clustering	partition	procedural
Hartigan (1972)	nested clustering	partition	deterministic
Hartigan (1975) two-way splitting	nested clustering	nested clustering (unrestricted)	procedural
Mirkin et al. (1995)	nested clustering	nested clustering (restricted)	procedural
De Soete and Carroll (1996)	nested clustering	nested clustering (restricted)	deterministic
Eckes and Orlik (1993)	overlapping clustering	overlapping clustering	procedural
DeSarbo (1982)	overlapping clustering	overlapping clustering	deterministic (simple additive)
Mickey et al. (1983)	overlapping clustering	overlapping clustering	deterministic (boolean)

Figure 4.4: Schematic overview of Van Mechelen, Bock and De Boeck (2004)

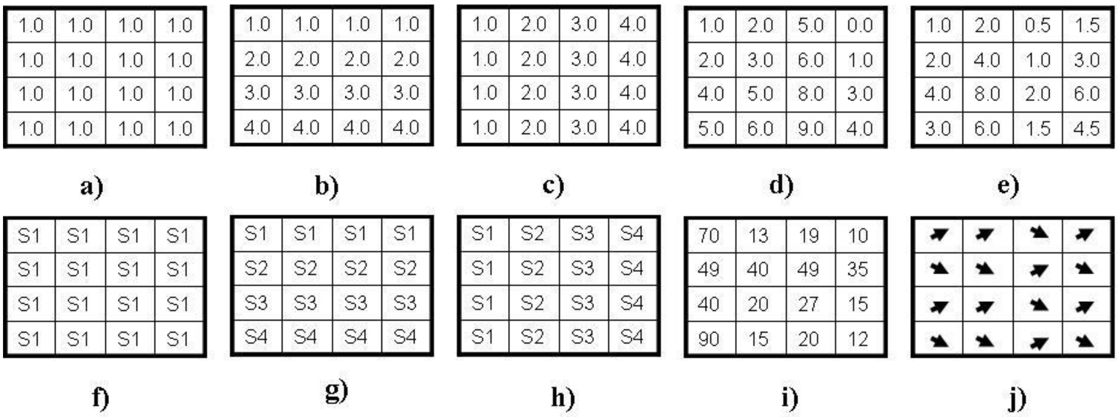
4.3 Madeira and Oliveira (2004) taxonomy

The second review classifies double clustering algorithms according to four principles:

- the type of blocks they can find. The authors identify four major classes (Figure 4.5):

  1. blocks with *constant values*;
  2. blocks with *constant rows or columns* values;
  3. blocks with *coherent values*;
  4. blocks with *coherent evolutions.*

Figure 4.5: Examples of different types of blocks: a) constant block; b) constant rows; c) constant columns; d) coherent values (additive model); e) coherent values (multiplicative model); f) overall coherent evolution; g) coherent evolution on the rows; h) coherent evolution on the columns; i) coherent evolution on the columns; j) coherent sign changes on rows and columns



a)

1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0

b)

1.0	1.0	1.0	1.0
2.0	2.0	2.0	2.0
3.0	3.0	3.0	3.0
4.0	4.0	4.0	4.0

c)

1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0

d)

1.0	2.0	5.0	0.0
2.0	3.0	6.0	1.0
4.0	5.0	8.0	3.0
5.0	6.0	9.0	4.0

e)

1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

f)

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

g)

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

h)

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

i)

70	13	19	10
49	40	49	35
40	20	27	15
90	15	20	12

j)

↗	↗	↘	↗
↘	↘	↗	↘
↗	↗	↘	↗
↘	↘	↗	↘

- The way multiple blocks are treated and the associated block structure. Some algorithms find only one cluster, while other assume the existence of several blocks. In the second case, we can obtain the following blocks structures (Figure 4.6):

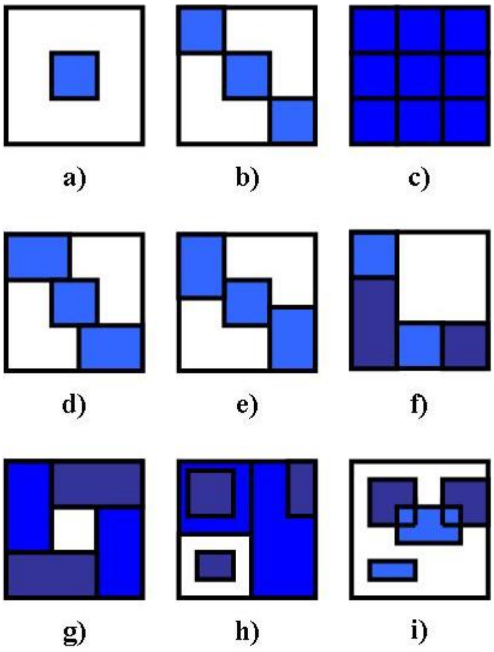
  1. exclusive row and column blocks (rectangular diagonal blocks after row and column reorder);
  2. non-overlapping blocks with checkerboard structure;
  3. exclusive-rows blocks;
  4. exclusive-columns blocks;
  5. non-overlapping blocks with tree structure;
  6. non-overlapping non-exclusive blocks;
  7. overlapping blocks with hierarchical structure;
  8. arbitrarily positioned overlapping blocks.

- The specific algorithm used to identify each block. They divided the algorithms into five sets:

  1. iterative row and column clustering combination;
  2. divide and conquer;
  3. greedy iterative search;
  4. exhaustive block enumeration;
  5. distribution parameter identification.

  The conceptually simpler way to perform double clustering using existing techniques is to apply standard clustering methods on rows and columns separately, successively combining the results using some kind

Figure 4.6: Examples of different block structures: a) single block; b) exclusive row and column blocks; c) checkerboard structure; d) exclusive-rows blocks; e) exclusive-columns blocks; f) non-overlapping blocks with tree structure; g) non-overlapping non-exclusive blocks; h) overlapping blocks with hierarchical structure; i) arbitrarily positioned overlapping blocks



of iterative procedure to obtain blocks. These approaches do not evaluate the quality of the resulting double clustering directly, but separately on each dimension (Getz et al., 2000; Tang et al., 2001; Busygin et al., 2002).

Divide-and-conquer algorithms break the problem onto several sub-problems that are similar to the original problem and are characterized by a smaller size; they solve the sub-problems recursively and then combine the sub-solutions to create an overall solution to the original problem. An historical example is given by Hartigan (1972). He proposed a partition based on *direct clustering* algorithm splitting the original data matrix into a set of blocks, where each block has constant values. The loss function used to evaluate constant blocks is the within-blocks variance.

Greedy iterative search methods are based on the idea of creating blocks by adding or removing rows/columns from them, maximizing a local choice hoping that this choice will lead to a globally optimal solution (Cheng and Church, 2000; Califano et al., 2000; Yang et al., 2002; Ben-Dor et al., 2002; Yang et al., 2003; Klugar et al., 2003; Murali and Kasif, 2003; Cho et al., 2004).

Exhaustive block enumeration methods are based on the idea that the best blocks can only be identified using an exhaustive enumeration of all possible blocks in the data matrix (Wang et al., 2002; Tanay et al., 2002; Liu and Wang, 2003).

Most of this approaches evaluate the quality of the solution by analyzing the values of the loss function (Hartigan, 1972; Cheng and Church, 2000; Lazzeroni and Owen, 2002; Wang et al., 2002; Yang et al., 2002; Yang et al., 2003; Cho et al., 2004); when an explicit statistical model underlies the loss function a formal approach can be used (Segal et al., 2001; Segal et al., 2003; Sheng et al., 2003; Murali and Kasif, 2003), by applying standard hypotheses testing (Califano et al., 2000; Tanay et al., 2002; Ben-Dor et al., 2002; Klugar et al., 2003).

- the domain of application of each algorithm (biological, information retrieval and text mining, market research, etc).

Figure 4.7 presents a summary of the different algorithms according to the different dimensions (of analysis) considered by Madeira and Oliveira (2004).

Authors	**Type**	**Structure**	**Discovery**	**Approach**
Hartigan (1972)	constant	nonoverlapping blocks with tree structure	one set at a time	divide and conquer
Cheng and Church (2000)	coherent values	arbitrarily positioned overlapping blocks	one at a time	greedy
Yang et al. (2002)	coherent values	arbitrarily positioned overlapping blocks	simultaneous	greedy
Yang et al. (2003)	coherent values	arbitrarily positioned overlapping blocks	simultaneous	greedy
Wang et al. (2002)	coherent values	exclusive columns blocks	simultaneous	exhaustive block enumeration
Lazzeroni and Owen (2002)	coherent values	arbitrarily positioned overlapping blocks	one at a time	distribution parameter identification
Segal et al. (2001)	constant columns	exclusive row and column blocks	simultaneous	distribution parameter identification
Segal et al. (2003)	coherent values	arbitrarily positioned overlapping blocks	simultaneous	distribution parameter identification
Getz et al. (2000)	coherent values	arbitrarily positioned overlapping blocks	one set at a time	iterative row and column clustering combination
Tang et al. (2001)	coherent values	exclusive rows/ columns blocks	one set at a time	iterative row and column clustering combination
Busygin et al. (2002)	coherent values	exclusive row and column blocks/ checkerboard structure	simultaneous	iterative row and column clustering combination
Califano et al. (2000)	constant rows	arbitrarily positioned overlapping blocks	simultaneous	greedy
Klugar et al. (2003)	coherent values	checkerboard structure	simultaneous	greedy
Sheng et al. (2003)	constant columns	exclusive rows/ columns blocks	one at a time	distribution parameter identification
Ben-Dor et al. (2002)	coherent evolution	single block/ arbitrarily positioned overlapping blocks	one at a time	greedy
Tanay et al. (2002)	coherent evolution	arbitrarily positioned overlapping blocks	simultaneous	exhaustive block enumeration
Murali and Kasif (2003)	coherent evolution	single block/ arbitrarily positioned overlapping blocks	simultaneous	greedy
Liu and Wang (2003)	coherent evolution	arbitrarily positioned overlapping blocks	simultaneous	exhaustive block enumeration
Cho et al. (2004)	constant/coherent values	arbitrarily positioned overlapping blocks	simultaneous	greedy

Figure 4.7: Schematic overview of Madeira and Oliveira (2004)

4.4 Conclusions

Besides the methods discussed in this Chapter, other approaches to double clustering have been introduced. We have tried to provide a framework for a better understanding of common as well as of distinctive features of double clustering methods. Also, the reviews may provide a bridge to transfer ideas and tools developed for one area of study to other areas.

In recent conferences (IFCS-2006 and COMPSTAT-2006), Van Mechelen (2006) and Van Mechelen and Schepers (2006) have proposed to extend the reviews on double clustering methods by focusing on:

- set-theoretical nature of clusters (partition, nested and/or overlapping clustering);
- internal structure of data clusters: homogeneity, rows effects, columns effects, rows and columns effects (additive structure), perfect correlation between rows or between columns, perfect rank correlation between rows or between columns (single monotonicity);
- mathematical operators used to define overlapping clustering (sum, average, minimum, maximum, product, etc);

Van Mechelen (2006) stressed the idea that double clustering models are particular case of the following models:

$$x_{ij} = \sum_{k,q} u_{ik} \bar{x}_{kq} v_{jq} + e_{ij}$$

or

$$x_{ij} = \max_{k,q}(u_{ik} \bar{x}_{kq} v_{jq}) + e_{ij}$$

or in general,

$$\mathbf{X} = f(\mathbf{U}, \bar{\mathbf{X}}, \mathbf{V}) + \mathbf{E} \tag{4.4}$$

where x_{ij} is the generic entry of data matrix $\mathbf{X}$ to be clustered ($i = 1, ..., n$; $j = 1, ..., J$), u_{ik} is the generic entry of a rows membership matrix $\mathbf{U}$ ($i = 1, ..., n$; $k = 1, ..., K$), v_{jq} is the generic entry of a columns membership matrix $\mathbf{V}$ ($j = 1, ..., J$; $q = 1, ..., Q$). Finally, e_{ij} represents the generic entry of a residual matrix $\mathbf{E}$.

Different specifications of the set-theoretical nature of $\mathbf{U}$ and $\mathbf{V}$ (constrained, unconstrained, constant, stochastic, etc) identify different approaches to the double clustering task.

However, this general model does not include those computational procedures which are not based on an explicit optimization of an overall objective function and procedures that optimize a criterion other than the optimal “reconstruction” of the data matrix (such as finding a single best block, or finding that double clustering optimally preserving the dependence or interaction in the data). An extension of this general model can be made by including heterogeneity sources in the data blocks, models for multiway data and models that include a categorical reduction, a dimensional reduction, and possibly no reduction at all.

Chapter 5

New approaches to double clustering

In this Chapter, we will discuss the extension of two well-known standard clustering methodologies, namely the K-means and the finite mixture model, to simultaneous cluster units and variables. Model-based methods that will be discussed are quite general and can be applied to a large number of high-dimensional data clustering problems.

We will first describe the *double K-means* model introduced by Vichi (2000), where model parameters are estimated by using a least-squares approach. Here, we propose to estimate model parameters through a maximum likelihood approach and define three coordinate ascent algorithms to give an efficient solution to fitting model. It can be noticed that double K-means is a particular case of the general model (4.4) proposed by Van Mechelen (2006) and Van Mechelen and Schepers (2006), where $\mathbf{U}$ and $\mathbf{V}$ are binary and row stochastic membership matrices.

After, we propose a simple factorial representation of component-specific means in finite mixture models extending the work of Rocci and Vichi (2002) to cluster variables as well. Since component specific densities may be far from Gaussianity, we introduce a hierarchical extension of finite mixture models following the proposals of Vermunt (2003) and Li (2005). In this way, we achieve the further aim to define specific clusters of variables within each cluster of units. Both models are evaluated through simulation studies and will be discussed in Part II using benchmark gene expression data.

5.1 Double K-means

In this Section a clustering technique that allows to simultaneously cluster units and variables is presented. It is referred to the *double K-means* and represents an extension of standard K-means (McQueen, 1967; see Section 2.1) to simultaneously cluster objects and features Vichi (2000). He proposed to estimate model parameters using a lest-squares approach, optimizing a quadratic objective function subject to a set of constraints due to the required clustering structure (e.g. partitions, coverings or packing) and clustering type (hard or fuzzy). In this dissertation, we will focus on hard partitions; in particular, we propose to estimate the model parameters using a maximum likelihood approach (as in Martella and Vichi, 2006) and discuss the advantages of using this approach.

Before discussing the method we introduce some notation and terminology which will be used throughout the rest of this section.

Notational Preliminaries

n, J, K, Q number of: units, variables, clusters of units, clusters of variables;

$\mathbf{X}$ $(n \times J)$ observed matrix of J quantitative variables n units (objects, individuals);

$\bar{\mathbf{X}}$ $(K \times Q)$ unknown matrix of block units-variables centroids where the generic element $\bar{x}_{kq}$ is the expected profile of the k-th unit cluster and q-th variable cluster ($k = 1, ..., K$ and $q = 1, ..., Q$);

$\mathbf{U}(n \times K)$ binary and row stochastic matrix of unit cluster membership, where $u_{ik} = 1$ if the i-th unit belongs to cluster k, 0 otherwise;

$\mathbf{V}(J \times Q)$ binary and row stochastic matrix of variable cluster membership, where $v_{jq} = 1$ if the j-th variable belongs to cluster q, 0 otherwise;

$\mathbf{E}$ $(n \times J)$ residual component matrix.

Using the above notation, the double K-means model can be written as:

$$\mathbf{X} = \mathbf{U}\bar{\mathbf{X}}\mathbf{V}' + \mathbf{E}, \qquad (5.1)$$

where $\mathbf{U}$ and $\mathbf{V}$ are binary and row stochastic matrices identifying units and variables clusters, respectively. It can be noticed that when $\mathbf{V} = \mathbf{I}_J$ or $\mathbf{U} = \mathbf{I}_n$ the double K-means collapses to the ordinary K-means on units or on variables respectively.

In particular, it can be expressed in row form as

$$\mathbf{x}_i = \mathbf{V}\bar{\mathbf{X}}'\mathbf{u}_i + \mathbf{e}_i \qquad i = 1, ..., n, \qquad (5.2)$$

where $\mathbf{x}_i$ is a $(J \times 1)$ vector representing the i-th row of $\mathbf{X}$ and $\mathbf{u}_i$ is the i-th unit membership vector. Vector $\mathbf{e}_i$ represents the i-th row of $\mathbf{E}$.

In the same way, model (5.1) can be expressed in column form as

$$\mathbf{x}^j = \mathbf{U}\bar{\mathbf{X}}\mathbf{v}_j + \mathbf{e}_j \qquad j = 1, ..., J, \qquad (5.3)$$

where $\mathbf{x}^j$ is a $(n \times 1)$ vector representing the j-th column of $\mathbf{X}$ and $\mathbf{v}_j$ is the j-th variable membership vector. Vector $\mathbf{e}_j$ represents the j-th column of $\mathbf{E}$.

In the next Section, we will discuss the least-squares approach to parameter estimation (Section 5.1.1), while in Section 5.1.2 we will show how model parameters can be estimated through a maximum likelihood approach.

5.1.1 Least-squares approach

The problem of determining a block partition of a data matrix $\mathbf{X}$ can be formalized by considering the loss function

$$min_{\mathbf{U},\bar{\mathbf{X}},\mathbf{V}} \parallel \mathbf{X} - \mathbf{U}\bar{\mathbf{X}}\mathbf{V}' \parallel^2, \tag{5.4}$$

subject to constraints:

$u_{ik} \in \{0,1\}$, $i = 1, ..., n$; $k = 1, ..., K$;

$\sum_{k=1}^{K} u_{ik} = 1$, $i = 1, ..., n$;

$v_{jq} \in \{0,1\}$, $j = 1, ..., J$; $q = 1, ..., Q$,

$\sum_{q=1}^{Q} v_{jq} = 1$, $j = 1, ..., J$.

In other words, we require that each block consists of entries that are as much similar as possible in a least-squares sense. Let us suppose $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ are the corresponding estimates; in this case, problem (5.4) can be reduced to determinate the LS solution of a generalized multivariate regression problem

$$min_{\bar{\mathbf{X}}} \parallel \mathbf{X} - \hat{\mathbf{U}}\bar{\mathbf{X}}\hat{\mathbf{V}}' \parallel^2, \tag{5.5}$$

which leads to the solution

$$\hat{\bar{\mathbf{X}}} = (\hat{\mathbf{U}}'\hat{\mathbf{U}})^{-1}\hat{\mathbf{U}}'\mathbf{X}\hat{\mathbf{V}}(\hat{\mathbf{V}}'\hat{\mathbf{V}})^{-1}. \tag{5.6}$$

However, since **U** and **V** are unknown, Vichi (2000) proposed an Alternating Least-Squares (ALS) algorithm for parameter estimation. It sequentially and recursively solves assignment problems, as shown in Table 5.1. The ALS algorithm, at each step, monotonically decreases the loss function converging to a stationary point (local or global minimum). Initial values for **U** and **V** can be randomly chosen; obviously, using different random starting points for **U** and **V**, we can increase the chance of finding a global minimum.

Initialization.	Choose initial values for $\mathbf{U}$ and $\mathbf{V}$. Such values can be chosen randomly or in a rationale way.
Step 1: Update $\bar{\mathbf{X}}$.	Given the current estimates of $\mathbf{U}$ and $\mathbf{V}$, update $\bar{\mathbf{X}}$ using (5.6).
Step 2: Update $\mathbf{U}$.	Minimize (5.4) over $\mathbf{U}$, given the current estimate of $\mathbf{V}$ and $\bar{\mathbf{X}}$.
Step 3: Update $\mathbf{V}$.	Minimize (5.4) over $\mathbf{V}$, given the current estimate of $\mathbf{U}$ and $\bar{\mathbf{X}}$.
Stopping rule.	Function (5.4) is computed for the current values of $\mathbf{U}$, $\mathbf{V}$ and $\bar{\mathbf{X}}$. If the function is considerably lower than in the previous iteration, $\mathbf{U}$, $\mathbf{V}$ and $\bar{\mathbf{X}}$ are updated once more according to step 1, 2 and 3. Otherwise, the process has converged.

Table 5.1: Alternating least-squares algorithm for double K-means

5.1.2 Maximum likelihood approach

Let us consider the double K-means model written in row form (5.2) and assume that the mean vector and the covariance matrix of the random vector $\mathbf{x}_i$ can be written as:

- $\mathrm{E}(\mathbf{x}_i) = \mathbf{V}\bar{\mathbf{X}}'\mathbf{u}_i$, for $i = 1, ..., n$;
- $\mathrm{Var}(\mathbf{x}_i) = \mathbf{\Sigma}$, for $i = 1, ..., n$.

We further assume that

$$\mathbf{x}_i \sim \mathrm{MVN}(\mathbf{V}\bar{\mathbf{X}}'\mathbf{u}_i, \mathbf{\Sigma})$$

that is

$$f(\mathbf{x}_i, \boldsymbol{\phi}) = \frac{1}{\sqrt{(2\pi)^J}}|\mathbf{\Sigma}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{V}\bar{\mathbf{X}}'\mathbf{u}_i)'\mathbf{\Sigma}^{-1}(\mathbf{x}_i - \mathbf{V}\bar{\mathbf{X}}'\mathbf{u}_i)\right\},$$

where $\boldsymbol{\phi} = \{\bar{\mathbf{X}}, \mathbf{\Sigma}, \mathbf{U}, \mathbf{V}\}$ is the parameter set.

Let $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$ be a sample of i.i.d. elements drawn from the density $f(\mathbf{x}_i, \boldsymbol{\phi})$; the corresponding likelihood function is:

$$L(\mathbf{U}, \mathbf{V}, \bar{\mathbf{X}}, \mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^{Jn}}}|\mathbf{\Sigma}|^{-\frac{n}{2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{V}\bar{\mathbf{X}}'\mathbf{u}_i)'\mathbf{\Sigma}^{-1}(\mathbf{x}_i - \mathbf{V}\bar{\mathbf{X}}'\mathbf{u}_i)\right\} =$$

$$= \frac{1}{\sqrt{(2\pi)^{Jn}}}|\mathbf{\Sigma}^{-1}|^{\frac{n}{2}} \exp\left\{-\frac{1}{2}tr\left[\mathbf{\Sigma}^{-1}(\mathbf{X} - \mathbf{U}\bar{\mathbf{X}}\mathbf{V}')'(\mathbf{X} - \mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\right]\right\},$$

and the log-likelihood is

$$l = \ln L(\mathbf{U}, \mathbf{V}, \bar{\mathbf{X}}, \mathbf{\Sigma}) = -nJ \ln \sqrt{2\pi} + \frac{n}{2} \ln |\mathbf{\Sigma}^{-1}| - \frac{1}{2}tr\left[\mathbf{\Sigma}^{-1}(\mathbf{X} - \mathbf{U}\bar{\mathbf{X}}\mathbf{V}')'(\mathbf{X} - \mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\right].$$

ML estimation of model parameters is obtained by maximizing the log-likelihood function, l, with respect to $\mathbf{U}$, $\mathbf{V}$, $\bar{\mathbf{X}}$ and $\mathbf{\Sigma}$ subject to binary and row stochastic constraints on $\mathbf{U}$ and $\mathbf{V}$.

Fixed $\mathbf{U}$ and $\mathbf{V}$, the optimal $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$ are the maximum likelihood solutions of the (generalized) multivariate regression problem $\mathbf{X} = \mathbf{U}\bar{\mathbf{X}}\mathbf{V}' + \mathbf{E}$, where $\mathbf{E}$ is a residual term.

However, since $\mathbf{U}$ and $\mathbf{V}$ are unknown, the MLEs of the double K-means model can be obtained by algorithms that alternatively maximize the log-likelihood function.

The fundamental steps of these algorithms can be described as follows.

- **Updating $\bar{\mathbf{X}}$:**

  When $\mathbf{U}$ and $\mathbf{V}$ are fixed, the ML estimate of $\bar{\mathbf{X}}$ is given by solving the corresponding likelihood equations; thus, we have

$$\frac{\partial l}{\partial \bar{\mathbf{X}}} \propto \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{-\frac{1}{2}tr\Big[\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')'(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\Big]\right\} =$$

$$= \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{-\frac{1}{2}tr\Big[\boldsymbol{\Sigma}^{-1}(\mathbf{X}'-\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}')(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\Big]\right\} =$$

$$= \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{-\frac{1}{2}tr\Big[\boldsymbol{\Sigma}^{-1}(\mathbf{X}'\mathbf{X}-2\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{X}+\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\Big]\right\} =$$

$$= \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{-\frac{1}{2}\Big[tr\Big[\boldsymbol{\Sigma}^{-1}(\mathbf{X}'\mathbf{X}-2\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{X})\Big]+tr\Big[\boldsymbol{\Sigma}^{-1}(\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\Big]\Big]\right\} = 0.$$

  Let us suppose

$$A = -\frac{1}{2}tr\Big[\boldsymbol{\Sigma}^{-1}(\mathbf{X}'\mathbf{X}-2\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{X})\Big]$$

  and

$$B = -\frac{1}{2}tr\Big[\boldsymbol{\Sigma}^{-1}(\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\Big].$$

  We consider $\frac{\partial A}{\partial \bar{\mathbf{X}}}$:

$$\frac{\partial A}{\partial \bar{\mathbf{X}}} = \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{-\frac{1}{2}tr\Big[\mathbf{\Sigma}^{-1}(\mathbf{X}'\mathbf{X} - 2\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{X})\Big]\right\} =$$

$$= \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{-\frac{1}{2}tr(\mathbf{\Sigma}^{-1}\mathbf{X}'\mathbf{X}) + \frac{1}{2}tr(2\mathbf{\Sigma}^{-1}\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{X})\right\} =$$

$$= \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{-\frac{1}{2}tr(\mathbf{\Sigma}^{-1}\mathbf{X}'\mathbf{X})\right\} + \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{\frac{1}{2}tr(2\mathbf{\Sigma}^{-1}\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{X})\right\} =$$

$$= \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{\frac{1}{2}tr(2\mathbf{\Sigma}^{-1}\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{X})\right\} = \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{tr(\mathbf{\Sigma}^{-1}\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{X})\right\} =$$

$$= \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{tr(\mathbf{U}'\mathbf{X}\mathbf{\Sigma}^{-1}\mathbf{V}\bar{\mathbf{X}}')\right\} = \mathbf{U}'\mathbf{X}\mathbf{\Sigma}^{-1}\mathbf{V},$$

and $\frac{\partial B}{\partial \bar{\mathbf{X}}}$:

$$\frac{\partial B}{\partial \bar{\mathbf{X}}} = \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{-\frac{1}{2}tr\Big[\mathbf{\Sigma}^{-1}(\mathbf{V}\bar{\mathbf{X}}'\mathbf{U}'\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\Big]\right\} =$$

$$= \frac{\partial}{\partial \bar{\mathbf{X}}}\left\{-\frac{1}{2}tr\Big[\mathbf{U}'\mathbf{U}\bar{\mathbf{X}}\mathbf{V}'\mathbf{\Sigma}^{-1}\mathbf{V}\bar{\mathbf{X}}'\Big]\right\} =$$

$$= -\frac{1}{2}\Big[(\mathbf{U}'\mathbf{U})'\bar{\mathbf{X}}(\mathbf{V}'\mathbf{\Sigma}^{-1}\mathbf{V})' + \mathbf{U}'\mathbf{U}\bar{\mathbf{X}}\mathbf{V}'\mathbf{\Sigma}^{-1}\mathbf{V}\Big] =$$

$$= -\frac{1}{2}\Big[2\mathbf{U}'\mathbf{U}\bar{\mathbf{X}}\mathbf{V}'\mathbf{\Sigma}^{-1}\mathbf{V}\Big] = -\mathbf{U}'\mathbf{U}\bar{\mathbf{X}}\mathbf{V}'\mathbf{\Sigma}^{-1}\mathbf{V}.$$

Thus,

$$\frac{\partial l}{\partial \bar{\mathbf{X}}} \propto \frac{\partial A}{\partial \bar{\mathbf{X}}} + \frac{\partial B}{\partial \bar{\mathbf{X}}} = \mathbf{U}'\mathbf{X}\mathbf{\Sigma}^{-1}\mathbf{V} - \mathbf{U}'\mathbf{U}\bar{\mathbf{X}}\mathbf{V}'\mathbf{\Sigma}^{-1}\mathbf{V} = 0$$

that is

$$\bar{\mathbf{X}}\mathbf{V}'\mathbf{\Sigma}^{-1}\mathbf{V} = (\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\mathbf{X}\mathbf{\Sigma}^{-1}\mathbf{V}.$$

Therefore, the ML estimator of $\bar{\mathbf{X}}$ is given by

$$\hat{\bar{\mathbf{X}}} = (\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\mathbf{X}\boldsymbol{\Sigma}^{-1}\mathbf{V}(\mathbf{V}'\boldsymbol{\Sigma}^{-1}\mathbf{V})^{-1} \tag{5.7}$$

- **Updating $\boldsymbol{\Sigma}$**:

When $\mathbf{U}$ and $\mathbf{V}$ are fixed, the ML estimate of $\boldsymbol{\Sigma}$ is given by solving the corresponding likelihood equations; thus, we have

$$\frac{\partial l}{\partial \boldsymbol{\Sigma}^{-1}} \propto \frac{\partial}{\partial \boldsymbol{\Sigma}^{-1}}\left\{\frac{n}{2}\ln|\boldsymbol{\Sigma}^{-1}| - \frac{1}{2}tr\left[\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')'(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\right]\right\} =$$

$$= \frac{\partial}{\partial \boldsymbol{\Sigma}^{-1}}\left\{\frac{n}{2}\ln|\boldsymbol{\Sigma}^{-1}|\right\} - \frac{\partial}{\partial \boldsymbol{\Sigma}^{-1}}\left\{\frac{1}{2}tr\left[\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')'(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')\right]\right\} =$$

$$= \frac{n}{2}\boldsymbol{\Sigma} - \frac{1}{2}(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')'(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}') = 0.$$

The ML estimator of $\boldsymbol{\Sigma}$ is given by

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n}(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}')'(\mathbf{X}-\mathbf{U}\bar{\mathbf{X}}\mathbf{V}'). \tag{5.8}$$

As far as the updating of $\mathbf{U}$ and $\mathbf{V}$ is concerned, the estimation is achieved row by row; that is, by putting the value 1 in the column position where the complete log-likelihood is maximized. In formulas:

- **Updating $\mathbf{U}$**:

When $\mathbf{V}$, $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$ are fixed, for each $i = 1, ..., n$ let

$$u_{ik} = \begin{cases} 1 & \text{if} \quad l(\cdot, u_{ik} = 1) = \max_h l(\cdot, u_{ih} = 1) \quad h = 1, ..., K; \quad h \neq k \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

- **Updating $\mathbf{V}$**:

When $\mathbf{U}$, $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$ are fixed, for each $j = 1, ..., J$ let

$$v_{jq} = \begin{cases} 1 & \text{if} \quad l(\cdot, v_{jq} = 1) = \max_z l(\cdot, v_{jz} = 1) \quad z = 1, ..., Q; \quad z \neq q \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

5.1.3 Coordinate ascent algorithms

In this Section we will propose three algorithms for parameter estimations in double K-means model under a maximum likelihood approach. In particular, those algorithms do coordinate ascent on the log-likelihood function. In fact, they pick a block of parameters and optimize the log-likelihood function over this block, considering all other parameters as fixed.

Algorithm 1

Initialization. Some initial values are chosen for $\mathbf{U}$ and $\mathbf{V}$. Such values can be chosen randomly or in a rationale way (e.g., using an alternating least-squares algorithm for double K-means, Vichi, 2000).

Step 1: Update $\mathbf{U}$, $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$. Maximization of $l(\cdot)$ with $\mathbf{V}$ fixed, by updating $\bar{\mathbf{X}}$ as in (5.7), $\boldsymbol{\Sigma}$ as in (5.8) and $\mathbf{U}$ as in (5.9).

Step 2: Update $\mathbf{V}$, $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$. Maximization of $l(\cdot)$ with $\mathbf{U}$ fixed, by updating $\bar{\mathbf{X}}$ as in (5.7), $\boldsymbol{\Sigma}$ as in (5.8) and $\mathbf{V}$ as in (5.10).

Stopping rule. The function value $l(\cdot)$ is computed for the current values of $\mathbf{U}$, $\mathbf{V}$, $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$. If the likelihood difference is above a specific threshold, then $\mathbf{U}$, $\mathbf{V}$, $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$ are updated once more according to Steps 1 and 2. Otherwise, the process has converged.

Algorithm 2

Initialization. Initial values are chosen for $\mathbf{U}$ and $\mathbf{V}$. Previously described approaches can be used.

Step 1: Update $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$. Given the current values for $\mathbf{U}$, $\mathbf{V}$, update $\bar{\mathbf{X}}$ using (5.7) and $\boldsymbol{\Sigma}$ using (5.8).

Step 2: Update $\mathbf{U}$ using (5.9), given the current estimate for $\mathbf{V}$, $\bar{\mathbf{X}}$, $\boldsymbol{\Sigma}$.

Step 3: Update $\mathbf{V}$ using (5.10), given the current estimate for $\mathbf{U}$, $\bar{\mathbf{X}}$, $\boldsymbol{\Sigma}$.

Stopping rule. The function value $l(\cdot)$ is computed for the current values of $\mathbf{U}$, $\mathbf{V}$, $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$. If the likelihood difference is increased above a specific threshold, $\mathbf{U}$, $\mathbf{V}$, $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$ are updated once more according to Steps 1, 2 and 3. Otherwise, the process has converged.

At each step of the algorithms (1 or 2), the log-likelihood function is monotonically increased and, since $l(\cdot)$ is bounded from above, it will converge to a stationary point which usually turns out to be at least a local maximum. To increase the chance of finding the global maximum, standard practice suggests to run the algorithm several times starting from different initial values for $\mathbf{U}$ and $\mathbf{V}$, retaining the best solution in terms of maximized log-likelihood or penalized criteria values.

It must be noticed that the first algorithm updates $\bar{\mathbf{X}}$ and $\boldsymbol{\Sigma}$ during the updating of $\mathbf{U}$ and $\mathbf{V}$ in order to improve these estimates; this implies an increase in computational complexity. In order to test if the performances of the first algorithm justify its computational complexity with respect to the second algorithm, a simulation study has been carried out. As it is showed in

Table (5.2), no considerable improvement can be registered for algorithm 1 VS algorithm 2; however, a meaningful decrease in computational complexity (e.g. CPU time) has been registered for the second algorithm.

Algorithm 2rid

Since in most practical cases the number of units is very large, we can think to speed up the algorithm by reducing the dimensionality of the units space, working on sub-matrices having K rows instead of n, $K < n$ (see Rocci and Vichi, 2004).

In fact, we rewrite the membership matrix $\mathbf{U}$ as $\tilde{\mathbf{U}}$=$\mathbf{U}L_{\mathbf{u}}^{-1}$, where $L_{\mathbf{u}} = (\mathbf{U}'\mathbf{U})^{1/2}$, i.e. a diagonal matrix having on the main diagonal the square roots of cluster cardinalities; it results that $\tilde{\mathbf{U}}'\tilde{\mathbf{U}} = \mathbf{n}$. Therefore, during the updating of $\mathbf{V}$ in Step 3 of algorithm 2 the function to be maximized can be decomposed as follows:

$$\begin{aligned} l(\cdot) \propto \| \boldsymbol{\Sigma}^{-1/2}(\mathbf{X} - \tilde{\mathbf{U}}L_{\mathbf{u}}\bar{\mathbf{X}}\mathbf{V}')' \|^2 = \| \boldsymbol{\Sigma}^{-1/2}\mathbf{X}' - \boldsymbol{\Sigma}^{-1/2}\mathbf{V}\bar{\mathbf{X}}'L_{\mathbf{u}}\tilde{\mathbf{U}}' \|^2 = \\ = \| \boldsymbol{\Sigma}^{-1/2}\mathbf{X}' \|^2 + \| \boldsymbol{\Sigma}^{-1/2}\mathbf{V}\bar{\mathbf{X}}'L_{\mathbf{u}}\tilde{\mathbf{U}}' \|^2 - 2tr(\boldsymbol{\Sigma}^{-1/2}\mathbf{X}'\tilde{\mathbf{U}}L_{\mathbf{u}}\bar{\mathbf{X}}\mathbf{V}'\boldsymbol{\Sigma}^{-1/2}) = \\ = \| \boldsymbol{\Sigma}^{-1/2}\mathbf{X}' \|^2 + \| \tilde{\mathbf{U}}'\mathbf{X}\boldsymbol{\Sigma}^{-1/2} \|^2 - \| \tilde{\mathbf{U}}'\mathbf{X}\boldsymbol{\Sigma}^{-1/2} \|^2 + \\ + \| \boldsymbol{\Sigma}^{-1/2}\mathbf{V}\bar{\mathbf{X}}'L_{\mathbf{u}}\tilde{\mathbf{U}}' \|^2 - 2tr(\tilde{\mathbf{U}}'\mathbf{X}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}^{-1/2}\mathbf{V}\bar{\mathbf{X}}'L_{\mathbf{u}}) = \\ = \| \boldsymbol{\Sigma}^{-1/2}\mathbf{X}' \|^2 - \| \tilde{\mathbf{U}}'\mathbf{X}\boldsymbol{\Sigma}^{-1/2} \|^2 + \| \boldsymbol{\Sigma}^{-1/2}(\tilde{\mathbf{U}}'\mathbf{X} - L_{\mathbf{u}}\bar{\mathbf{X}}\mathbf{V}')' \|^2 . \end{aligned} \tag{5.11}$$

Thus, we can proceed by maximizing only the third term. As shown in Table 5.2, we obtain the same solution obtained using algorithm 2, but with a substantial decrease in computational complexity.

Type Algorithm	Average Mrand for units partition	Average Mrand for variables partition	% of unit partitions= to the true partitions	% of variable partitions= to the true partitions	Average number of iterations	CPU time
1	0.87206	1	76	100	2.44	18416.4
2	0.94159	1	72	100	2.16	6512.33
2rid	0.94159	1	72	100	2.16	4529.34

Table 5.2: Performance results: K=3, Q=2

5.1.4 Conclusions

The double K-means model introduced by Vichi (2000) has been applied to simultaneous clustering of units and variables. Vichi (2000) proposed to estimate model parameters by using a LS approach while we have proposed to use a ML approach; we have also developed three coordinate ascent algorithms and tested their performance in a simulation study (see Part II of the dissertation). It can be observed that LS and ML approaches provide different optimal solutions. The substantial difference between LS and ML solutions lies behind the different allocations of units and variables to clusters. In fact, the LS method is based on the Euclidean distance while the ML method uses the Mahalanobis distance; the latter is a weighted Euclidean distance where weighting is expressed by the covariance matrix, which accounts for correlations in the analyzed data.

In others words, in the LS approach the covariance matrix is represented by the identity matrix $\mathbf{\Sigma} = \mathbf{I}_J$, i.e. the clusters are constrained to have a spherical orthogonal shape. In the ML approach, instead, clusters may have general shapes, e.g. elliptical. Moreover, model-based double K-means leads to define a set of criteria to choose the number of clusters which are based on a penalization of the log-likelihood function (e.g. BIC, AIC, etc). However, a limitation of this methodology is that variables clusters are fixed for all units clusters and the allocations of units and variables are made in an exclusive way.

5.2 Double Gaussian mixture model

The double K-means model with the following assumption

$$\mathbf{x}_i \sim \mathrm{MVN}(\mathbf{V}\bar{\mathbf{X}}'\mathbf{u}_i, \mathbf{\Sigma}), \qquad i = 1, ..., n$$

can be seen as a particular case of the Gaussian mixture model with component-specific mean vectors described by in Rocci and Vichi (2002). The proposed model, termed *double Gaussian mixture model*, is defined as follows:

$$f(\mathbf{y}_i; \boldsymbol{\phi}) = \sum_{k=1}^{K} \pi_k \varphi(\mathbf{y}_i; \mathbf{V}\tilde{\mathbf{u}}_k, \mathbf{\Sigma}_k),$$

where $\mathbf{V}$ $(J \times Q)$ is not a binary and row stochastic matrix of variable cluster membership, where $v_{jq} = 1$ if j-th variable belongs to cluster q, 0 otherwise (for $q = 1, ..., Q$ and $j = 1, ..., J$); $\tilde{\mathbf{u}}_k$ represents the deviation of the component-specific mean vector projected onto a Q-dimensional space ($Q < J$ and $k = 1, ..., K$) from the overall mean. The analogy between double K-means and double Gaussian mixture models is demonstrated by the following relationship:

$$\bar{\mathbf{X}}'\mathbf{u}_i = \tilde{\mathbf{u}}_k,$$

by considering exclusive allocations for units and constant component-specific covariance matrices ($\mathbf{\Sigma}_k = \mathbf{\Sigma}$, $k = 1, ..., K$) in the double Gaussian mixture model.

Thus, through this simple reparameterization of the component-specific mean vector, we are able to cluster both units and variables. In fact, the i-th unit is allocated to cluster k corresponding to the largest posterior probability, w_{ik} $(k = 1, ..., K)$; while the j-th variable is allocated to the q-th cluster through the matrix membership $\mathbf{V}$ $(q = 1, ..., Q)$.

However, the double Gaussian mixture model, as well as the double K-means model, specifies for each unit cluster the same variables partition and for each variable cluster the same units partition and, as well as the standard Gaussian mixture model, relies on the hypothesis that each group is drawn from a Gaussian density. This last assumption could be violated; in fact, an isolated group with a non-Gaussian distribution might be modelled by several mixture components, and the resulting components would no longer represent a cluster.

In the next Section, we introduce a hierarchical extension of the double Gaussian mixture model that combines the advantages of dealing with non-Gaussianity of component-specific densities and to define clusters of variables, where conditionally a different partition of the variables is allowed within each units cluster.

5.3 The hierarchical mixture model

In this Section, we introduce a clustering approach based on a hierarchical structure, allowing for both dependence within clusters and simultaneous clustering of units and variables (as described in Alfó, Martella and Vichi, 2006). The proposed approach extends the multilevel latent class model proposed by Vermunt (2003) and Li (2005) to two-way continuous data. Thanks to the hierarchical structure we distinguish clusters (2nd level) from components (1st level) allowing for greater flexibility in the shape of cluster specific densities represented as a finite mixture of Gaussian distributions. In order to cluster variables, we introduce a binary row stochastic matrix representing variables membership (as in double K-means, Vichi, 2000), by using a reparameterization of the component-specific mean vectors proposed by Rocci and Vichi (2002).

Notational Preliminaries

To discuss the hierarchical extension of the mixture model, we need to extend the adopted notation considering an (unobserved) extra-level.

n, J: number of units and variables;

K: number of 2nd level component densities (clusters);

T_k: number of 1st level component densities (components) within the k-th 2nd level cluster;

n_k: number of units within the k-th cluster ($k = 1, ..., K$);

n_{tk}: number of units within the t-th component in the k-th cluster ($k = 1, ..., K$; $t = 1, ..., T_K$);

Q_k: number of variable clusters within the k-th cluster ($k = 1, ..., K$).

Let $\mathbf{y}_i$ be a J-dimensional observation ($i = 1, ..., n$). We assume $\mathbf{y}_i$ is drawn from one of K 2nd level components, called clusters; each of these clusters is composed by T_k components. In details, the marginal density can be written as:

$$f(\mathbf{y}_i|\boldsymbol{\phi}) = \sum_{k=1}^{K} \pi_k f(\mathbf{y}_i|\boldsymbol{\theta}_k) \tag{5.12}$$

where $\boldsymbol{\phi} = (\pi_1, ...\pi_{K-1}, \boldsymbol{\theta}_1, ...\boldsymbol{\theta}_K)$, $\boldsymbol{\theta}_k$ is the k-th cluster specific parameter vector and π_k, $\sum_{k=1}^{K} \pi_k = 1$, is the prior probability that the observation $\mathbf{y}_i$ belongs to the k-th cluster $(k = 1, ..., K)$. The 2nd level describes units, $\mathbf{y}_i$'s, belonging to the same k-th cluster ignoring the T_k components; the 1st level describes units within each component, t, in the k-th cluster. The component specific density in the k-th cluster is assumed to be equal to:

$$f(\mathbf{y}_i|\boldsymbol{\theta}_k) = \sum_{t=1}^{T_k} \pi_{t|k} f(\mathbf{y}_i|\boldsymbol{\theta}_{t|k}), \tag{5.13}$$

where $\pi_{t|k} = Pr(i \in t | i \in k)$, $\sum_{t=1}^{T_k} \pi_{t|k} = 1$, is the conditional probability that the i-th observation $\mathbf{y}_i$ belongs to the t-th component within the k-th cluster $(k = 1, ..., K;\ t = 1, ..., T_k)$. It is worth noticing that if $\pi_k = 0$ then $\pi_{t|k} = 0$ for each $t = 1, ..., T_k$ $(k = 1, ..., K)$. In the following, we will assume J-variate Gaussian component specific densities $f(\mathbf{y}_i|\boldsymbol{\theta}_{t|k})$ with component-specific mean vectors and covariance matrices $\boldsymbol{\mu}_{t|k}$ and $\boldsymbol{\Sigma}_{t|k}$, i.e.

$$f(\mathbf{y}_i|\boldsymbol{\theta}_{t|k}) = \frac{1}{\sqrt{(2\pi)^J}} |\boldsymbol{\Sigma}_{t|k}|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(\mathbf{y}_i - \boldsymbol{\mu}_{t|k})' \boldsymbol{\Sigma}_{t|k}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_{t|k}) \right\}. \tag{5.14}$$

5.3.1 Maximum likelihood estimation

To extend the standard EM algorithm to the proposed hierarchical structure, we have to introduce the following component labels:

$$z_{ik} = \begin{cases} 1 & \text{if } \mathbf{y}_i \text{ belongs to the } k\text{-th (2nd level) cluster} \\ 0 & \text{otherwise} \end{cases} \tag{5.15}$$

and

$$z_{it|k} = \begin{cases} 1 & \text{if } \mathbf{y}_i \text{ belongs to the } t\text{-th (1st level) component} \\ & \text{within the } k\text{-th (2nd level) cluster} \\ 0 & \text{otherwise} \end{cases} \tag{5.16}$$

The $\mathbf{z}_i = (z_{i1}, ..., z_{iK})'$, for $i = 1, ..., n$, as well as the $\mathbf{z}_{i|k} = (z_{i1|k}, ..., z_{iT_k|k})'$ are assumed to come from Multinomial distributions with parameters π_k and, respectively, $\pi_{t|k}$. By treating these component labels as missing data, ML parameter estimation can be achieved by means of the EM algorithm. However, due to the high dimensionality of the estimation problem, we can not rely on a standard EM algorithm; rather we may turn to the *upward-downward* algorithm (Pearl, 1988).

Following expressions (5.12) to (5.14), the complete data log-likelihood function has the following form:

$$\log L_C(\boldsymbol{\phi}) = \log\left\{ \prod_{i=1}^{n} \left[\prod_{k=1}^{K} \pi_k \Big[\prod_{t=1}^{T_k} \pi_{t|k} f(\mathbf{y}_i|\boldsymbol{\theta}_{t|k}) \Big]^{z_{it|k}} \right]^{z_{ik}} \right\}$$

that is,

$$\begin{aligned} \log L_C(\boldsymbol{\phi}) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} \log(\pi_k) + \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T_k} z_{ik} z_{it|k} \log(\pi_{t|k}) + \\ + \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T_k} z_{ik} z_{it|k} \log\big[f(\mathbf{y}_i|\boldsymbol{\theta}_{t|k})\big] \end{aligned} \tag{5.17}$$

where $\boldsymbol{\phi}$ represents the unknown parameter vector. Let us start the EM algorithm as before; in the $(h+1)$-th iteration of the E-step we compute the expected value of the complete data log-likelihood function. As usual, the $\log L_C(\boldsymbol{\phi})$ is linear in the component labels and taking the expectation implies that the component labels z_{ik} and $z_{it|k}$ are replaced by their expected values

$w_{ik}^{(h)} = Pr(z_{ik} = 1|\mathbf{y}; \boldsymbol{\phi}^{(h)})$ and $w_{it|k}^{(h)} = Pr(z_{it|k} = 1|z_{ik} = 1, \mathbf{y}; \boldsymbol{\phi}^{(h)})$: these are the estimated posterior probabilities that the i-th observation belongs to the k-th (2nd level) cluster and, respectively, to the t-th (1st level) component within the k-th cluster, conditional on the observed data and the current parameter estimates. Therefore, we could write

$$Q(\boldsymbol{\phi}, \boldsymbol{\phi}^{(h)}) = E\big[\log L_C(\boldsymbol{\phi})\big] = \sum_{i=1}^{n}\sum_{k=1}^{K} w_{ik}^{(h)} \log(\pi_k) + \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T_k} w_{ik}^{(h)} w_{it|k}^{(h)} \log(\pi_{t|k}) + \\ + \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T_k} w_{ik}^{(h)} w_{it|k}^{(h)} \log\big[f(\mathbf{y}_i|\boldsymbol{\theta}_{t|k})\big], \tag{5.18}$$

where $w_{ik}^{(h)} w_{it|k}^{(h)} = Pr(z_{ik} = 1, z_{it|k} = 1|\mathbf{y}; \boldsymbol{\phi}^{(h)})$ is the expected value of the product $z_{ik}z_{it|k}$ in (5.17).

Since the expected value is linear in the missing component indicators, the E-step reduces to the computation of $w_{ik}^{(h)}$ and $w_{it|k}^{(h)}$ which are given by

$$w_{ik}^{(h)} = \frac{\hat{\pi}_k^{(h)} \sum_{t=1}^{T_k} \hat{\pi}_{t|k}^{(h)} f(\mathbf{y}_i|\hat{\boldsymbol{\theta}}_{t|k}^{(h)})}{\sum_{k=1}^{K} \hat{\pi}_k^{(h)} \sum_{t=1}^{T_k} \hat{\pi}_{t|k}^{(h)} f(\mathbf{y}_i|\hat{\boldsymbol{\theta}}_{t|k}^{(h)})} \tag{5.19}$$

and

$$w_{it|k}^{(h)} = \frac{\hat{\pi}_{t|k}^{(h)} f(\mathbf{y}_i|\hat{\boldsymbol{\theta}}_{t|k}^{(h)})}{\sum_{t=1}^{T_k} \hat{\pi}_{t|k}^{(h)} f(\mathbf{y}_i|\hat{\boldsymbol{\theta}}_{t|k}^{(h)})} \tag{5.20}$$

respectively. As it can be seen, for each unit $i = 1, ..., n$, we first compute $\hat{\pi}_{t|k}^{(h)} f(\mathbf{y}_i|\hat{\boldsymbol{\theta}}_{t|k}^{(h)})$ for the $(t,\ k)$ combination and then collapse these over t to obtain $\sum_{t=1}^{T_k} \hat{\pi}_{t|k}^{(h)} f(\mathbf{y}_i|\hat{\boldsymbol{\theta}}_{t|k}^{(h)})$, which amounts to marginalizing over the (1st level) component. Collapsing the $\sum_{t=1}^{T_k} \hat{\pi}_{t|k}^{(h)} f(\mathbf{y}_i|\hat{\boldsymbol{\theta}}_{t|k}^{(h)})$ over k we obtain the posterior probabilities for (2nd level) clusters.

Vermunt (2003) refers to these steps as the *upward* steps exploiting the analogy with the *forward-backward* algorithm in the context of ML estimation in hidden Markov models. In fact, information from the 1st level of the hierarchical structure is passed to the 2nd level one. The *downward* step, instead, involves the computation of the joint posteriors of z_{ik} and $z_{it|k}$, that is

$$w_{ik}^{(h)} w_{it|k}^{(h)} = Pr(z_{ik} = 1, z_{it|k} = 1 | \mathbf{y}; \boldsymbol{\phi}^{(h)})$$

$(t = 1, ..., T_k;\ k = 1, ..., K)$ which enter as weights in the expected $\log L_c(\cdot)$.

In the $(h+1)$-th iteration of M-step, model parameters are estimated by maximizing the expected complete data log-likelihood. To maximize this expression, we can maximize the term containing π_k, the term containing $\pi_{t|k}$ and the term containing $\boldsymbol{\theta}_{t|k}$ independently since the likelihood can be easily factorized.

To find the estimate for π_k, we introduce the Lagrange function with constraint $\sum_{k=1}^{K} \pi_k = 1$, and solve the following equation:

$$\frac{\partial Q}{\partial \pi_k} = \frac{\partial}{\partial \pi_k} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{K} w_{ik} \log(\pi_k) + \lambda \Big(\sum_{k=1}^{K} \pi_k - 1 \Big) \right\} = 0$$

that is

$$\sum_{i=1}^{n} \frac{1}{\pi_k} w_{ik} + \lambda = 0.$$

Summing over k, we get that $\lambda = -n$ resulting in the $(h+1)$-th iteration estimate:

$$\pi_k^{(h+1)} = \frac{\sum_{i=1}^{n} w_{ik}^{(h)}}{n}. \tag{5.21}$$

The same procedure is employed for $\pi_{t|k}$. We introduce the Lagrange function with constraint $\sum_{t=1}^{T_k} \pi_{t|k} = 1$, and solve the following equation:

$$\frac{\partial Q}{\partial \pi_{t|k}} = \frac{\partial}{\partial \pi_{t|k}} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{t=1}^{T_k} w_{ik} w_{it|k} \log(\pi_{t|k}) + \lambda \Big(\sum_{t=1}^{T_k} \pi_{t|k} - 1 \Big) \right\} = 0$$

or

$$\sum_{i=1}^{n}\sum_{k=1}^{K}\frac{1}{\pi_{t|k}}w_{ik}w_{it|k}+\lambda=\sum_{i=1}^{n}\frac{1}{\pi_{t|k}}w_{ik}w_{it|k}+\frac{\lambda}{K}=0$$

Summing over t, we get again $\lambda=-K\sum_{i=1}^{n}w_{ik}$ resulting in the $(h+1)$-th iteration estimate:

$$\pi_{t|k}^{(h+1)}=\frac{\sum_{i=1}^{n}w_{it|k}^{(h)}w_{ik}^{(h)}}{\sum_{i=1}^{n}w_{ik}^{(h)}}=\frac{\sum_{i=1}^{n}w_{it|k}^{(h)}w_{ik}^{(h)}}{n\pi_{k}^{(h+1)}}. \tag{5.22}$$

For given parametric specifications of $f(\cdot)$, it is possible to get closed expressions for $\boldsymbol{\theta}_{t|k}$ as functions of other parameters. For example, in the Gaussian case we can obtain a closed form estimate $\boldsymbol{\theta}_{t|k}^{(h+1)}=(\boldsymbol{\mu}_{t|k}^{(h+1)},\boldsymbol{\Sigma}_{t|k}^{(h+1)})$.

Taking the log of Equation (5.14), ignoring constant terms, and substituting into (5.18), we get:

$$\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T_k}w_{ik}w_{it|k}\log\Big(f(\mathbf{y}_i|\boldsymbol{\mu}_{t|k},\boldsymbol{\Sigma}_{t|k})\Big)=$$

$$=\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T_k}w_{ik}w_{it|k}\left[-\frac{1}{2}\log(|\boldsymbol{\Sigma}_{t|k}|)-\frac{1}{2}(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})'\boldsymbol{\Sigma}_{t|k}^{-1}(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})\right] \tag{5.23}$$

Taking the derivative with respect to $\boldsymbol{\mu}_{t|k}$ and solving the likelihood equation, we get:

$$\sum_{i=1}^{n}\sum_{k=1}^{K}w_{ik}w_{it|k}\boldsymbol{\Sigma}_{t|k}^{-1}(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})=0.$$

Solving for $\boldsymbol{\mu}_{t|k}$ we obtain the $(h+1)$-th iteration estimate:

$$\boldsymbol{\mu}_{t|k}^{(h+1)}=\frac{\sum_{i=1}^{n}w_{it|k}^{(h)}w_{ik}^{(h)}\mathbf{y}_i}{\sum_{i=1}^{n}w_{it|k}^{(h)}w_{ik}^{(h)}}. \tag{5.24}$$

To find $\boldsymbol{\Sigma}_{t|k}$ let us rewrite Equation (5.23) as

$$\sum_{k=1}^{K}\sum_{t=1}^{T_k}\left[\frac{1}{2}\log(|\mathbf{\Sigma}_{t|k}^{-1}|)\sum_{i=1}^{n} w_{ik}w_{it|k}-\frac{1}{2}\sum_{i=1}^{n} w_{ik}w_{it|k}tr\left(\mathbf{\Sigma}_{t|k}^{-1}(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})'\right)\right].$$

Taking the derivative with respect to $\mathbf{\Sigma}_{t|k}^{-1}$, we get:

$$\frac{1}{2}\sum_{i=1}^{n} w_{ik}w_{it|k}\Big(2\mathbf{\Sigma}_{t|k}-\text{diag}(\mathbf{\Sigma}_{t|k})\Big)-$$

$$-\frac{1}{2}\sum_{i=1}^{n} w_{ik}w_{it|k}\left[2(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})'-\text{diag}\Big((\mathbf{y}_i-\boldsymbol{\mu}_{t|k})(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})'\Big)\right]=$$

$$=\frac{1}{2}\sum_{i=1}^{n} w_{ik}w_{it|k}\left[2\Big(\mathbf{\Sigma}_{t|k}-(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})'\Big)-\text{diag}\Big((\mathbf{\Sigma}_{t|k}-(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})'\Big)\right] \tag{5.25}$$

Solving the corresponding likelihood equation we get

$$\sum_{i=1}^{n} w_{ik}w_{it|k}\Big(\mathbf{\Sigma}_{t|k}-(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})(\mathbf{y}_i-\boldsymbol{\mu}_{t|k})'\Big)=0$$

obtaining the $(h+1)$-th iteration estimate:

$$\mathbf{\Sigma}_{t|k}^{(h+1)}=\frac{\sum_{i=1}^{n} w_{it|k}^{(h)}w_{ik}^{(h)}(\mathbf{y}_i-\boldsymbol{\mu}_{t|k}^{(h+1)})(\mathbf{y}_i-\boldsymbol{\mu}_{t|k}^{(h+1)})'}{\sum_{i=1}^{n} w_{it|k}^{(h)}w_{ik}^{(h)}}. \tag{5.26}$$

As a by product, the upward-downward algorithm provides a (fuzzy) joint posterior matrix given by the product of 1st and 2nd level posterior probabilities of component and cluster membership. Therefore, we can cluster objects $i=1,...,n$ according to joint posteriors of component and cluster membership, using a Maximum a Posteriori (MAP) approach, that is:

$$z_{ik}z_{it|k}=\begin{cases}1 & \text{if } k=\text{argmax}_{k=1,...,K}w_{ik} \text{ and if } t=\text{argmax}_{t=1,...,T_k}w_{it|k}\\ 0 & \text{otherwise}\end{cases}.$$

5.3.2 The double Gaussian hierarchical mixture model

In this Section, we discuss a particular specification of the hierarchical model. We assume that (1st level) component-specific mean vectors can be written as follows:

$$\boldsymbol{\mu}_{t|k} = \boldsymbol{\mu}_k + \mathbf{V}_k \mathbf{u}_{t|k} \tag{5.27}$$

where $\boldsymbol{\mu}_k$ represents a J-dimensional (2nd level) cluster-specific mean vector, and $\mathbf{V}_k$ is a $(J \times Q_k)$ binary and row stochastic matrix of variable cluster membership $(k = 1, ..., K)$. Here $v_{jq_k} = 1$, if the j-th variable belongs to the q_k-th cluster, 0 otherwise; unknown latent factors $\mathbf{u}_{t|k}$ project the (1st level) component specific mean deviations $(\boldsymbol{\mu}_{t|k} - \boldsymbol{\mu}_k)$ onto a low dimensional space $(Q_k < J)$, $k = 1, ..., K$ and $t = 1, ..., T_k$.

For a given unit i belonging to the t-th (1st level) component within the k-th (2nd level) cluster, we can rewrite the data vector as:

$$\mathbf{y}_{i_{t|k}} = \boldsymbol{\mu}_k + \mathbf{V}_k \mathbf{u}_{t|k} + \mathbf{e}_{i_{t|k}} \tag{5.28}$$

where $\mathbf{e}_{i_{t|k}}$ represents an additive and residual term, $k = 1, ..., K$, $t = 1, ..., T_k$, $i = 1, ..., n$. As usual in linear modelling, we assume $\mathbf{e}_{i_{t|k}}$ are (Gaussian) random variates with zero mean and covariance matrix $\boldsymbol{\Sigma}_{t|k}$, $k = 1, ..., K$, $t = 1, ..., T_k$, $i = 1, ..., n$; further, we assume that $\mathbf{e}_{i_{t|k}}$ and $\mathbf{u}_{t|k}$ are independent. In other words, we assume that

$$E(\mathbf{e}_{i_{t|k}}) = \mathbf{0},$$

$$E(\mathbf{e}_{i_{t|k}}\ \mathbf{e}'_{i_{t|k}}) = \boldsymbol{\Sigma}_{t|k}$$

and

$$E(\mathbf{e}_{i_{t|k}}\ \mathbf{u}'_{t|k}) = \mathbf{0}$$

In this context, we can distinguish between two classes of model according to whether we consider the vector $\mathbf{u}_{t|k}$ to be random or not.

In both cases, we shall remind that $\mathbf{u}_{t|k}$ are hardly identifiable but this fact is not of central interest since we are mainly interested in estimating $\mathbf{V}_k$. The use of a nonrandom vector $\mathbf{u}_{t|k}$ poses problems of inference because the likelihood function may not have a maximum, and then the MLE may not exist (Anderson and Rubin, 1956). For this reason, we shall proceed to the estimation of $\mathbf{u}_{t|k}$, $k = 1, ..., K$ and $t = 1, ..., T_k$, assuming that the structural parameters $\mathbf{V}_k$, $\mathbf{\Sigma}_{t|k}$ and $\boldsymbol{\mu}_k$ are known.

5.3.3 Nonrandom $\mathbf{u}_{t|k}$

Let us assume $\mathbf{u}_{t|k}$, $k = 1, ..., K$ $t = 1, ..., T_k$, be nonrandom vectors. In this case, $\mathbf{y}_i^\star = (\mathbf{y}_{i_{t|k}} - \boldsymbol{\mu}_k)$ is drawn from a distribution with mean $\mathbf{V}_k\mathbf{u}_{t|k}$:

$$E(\mathbf{y}_i^\star) = E(\mathbf{V}_k\mathbf{u}_{t|k} + \mathbf{e}_{i_{t|k}}) = \mathbf{V}_k\mathbf{u}_{t|k} + E(\mathbf{e}_{i_{t|k}}) = \mathbf{V}_k\mathbf{u}_{t|k}$$

and covariance matrix:

$$E\Big[\big(\mathbf{y}_i^\star - E(\mathbf{y}_i^\star)\big)\big(\mathbf{y}_i^\star - E(\mathbf{y}_i^\star)\big)'\Big] = E\Big[\big(\mathbf{y}_i^\star - \mathbf{V}_k\mathbf{u}_{t|k}\big)\big(\mathbf{y}_i^\star - \mathbf{V}_k\mathbf{u}_{t|k}\big)'\Big] = \mathbf{\Sigma}_{t|k}$$

If we assume $\mathbf{e}_{i_{t|k}}$ are J-variate Gaussian random variables, we have:

$$\mathbf{y}_i^\star \sim \mathrm{MVN}(\mathbf{V}_k\mathbf{u}_{t|k}, \mathbf{\Sigma}_{t|k}). \tag{5.29}$$

We can obtain maximum likelihood estimates of $\mathbf{u}_{t|k}$ ($k = 1, ..., K$ $t = 1, ..., T_k$) as follows. The expected value of the complete data log-likelihood function is given by

$$Q(\boldsymbol{\phi}, \cdot) = E\big[\log L_c(\boldsymbol{\phi})\big] = \sum_{i=1}^{n}\sum_{k=1}^{K} w_{ik}\log(\pi_k) + \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T_k} w_{ik}w_{it|k}\log(\pi_{t|k}) + \\ + \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T_k} w_{ik}w_{it|k}\Big[-\frac{1}{2}\log|\mathbf{\Sigma}_{t|k}^{-1}| - \frac{1}{2}(\mathbf{y}_i^\star - \mathbf{V}_k\mathbf{u}_{t|k})'\mathbf{\Sigma}_{t|k}^{-1}(\mathbf{y}_i^\star - \mathbf{V}_k\mathbf{u}_{t|k})\Big] \tag{5.30}$$

Differentiating $Q(\boldsymbol{\phi}, \cdot)$ with respect to $\mathbf{u}_{t|k}$ and equating the derivates to zero, we obtain

$$\frac{\partial}{\partial \mathbf{u}_{t|k}} \left[\sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{t=1}^{T_k} w_{ik} w_{it|k} (\mathbf{y}_i^\star - \mathbf{V}_k \mathbf{u}_{t|k})' \boldsymbol{\Sigma}_{t|k}^{-1} (\mathbf{y}_i^\star - \mathbf{V}_k \mathbf{u}_{t|k}) \right] = 0$$

and thus

$$\hat{\mathbf{u}}_{t|k} = (\mathbf{V}_k' \boldsymbol{\Sigma}_{t|k}^{-1} \mathbf{V}_k)^{-1} \mathbf{V}_k' \boldsymbol{\Sigma}_{t|k}^{-1} (\sum_{i=1}^{n} w_{ik} w_{it|k} \mathbf{y}_i^\star)(\sum_{i=1}^{n} w_{ik} w_{it|k})^{-1} \tag{5.31}$$

$k = 1, ..., K$ and $t = 1, ..., T_k$.

It is worth noticing that the reparameterization of the mean vectors given in (5.27) has some links with Factor Analysis model, where $\mathbf{V}_k$ represents the factor loading matrix and $\mathbf{u}_{t|k}$ represent the factor scores. In this perspective, the proposed estimator $\hat{\mathbf{u}}_{t|k}$ detailed in (5.31), $k = 1, ..., K$ $t = 1, ..., T_k$, closely resembles the Bartlett estimator of factor scores based on the GLS approach (Bartlett, 1937; 1938).

5.3.4 Random $\mathbf{u}_{t|k}$

If we assume $\mathbf{u}_{t|k}$, $k = 1, ..., K$ $t = 1, ..., T_k$, are J-variate Gaussian random variables with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{I}_{Q_k}$ (i.e., the $\mathbf{u}_{t|k}$ are independent and homoscedastic, $k = 1, ..., K$ $t = 1, ..., T_k$), the joint Gaussian distribution of $(\mathbf{y}_{i_{t|k}}, \mathbf{u}_{t|k})$ can be written as

$$\begin{bmatrix} \mathbf{y}_{i_{t|k}} \\ \mathbf{u}_{t|k} \end{bmatrix} \sim MVN \left[\begin{pmatrix} \boldsymbol{\mu}_k \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{V}_k \mathbf{V}_k' + \boldsymbol{\Sigma}_{t|k} & \mathbf{V}_k \\ \mathbf{V}_k' & \mathbf{I}_{Q_k} \end{pmatrix} \right]. \tag{5.32}$$

To find a feasible estimator for $\mathbf{u}_{t|k}$, we use the posterior mean of $\mathbf{u}_{t|k}$ given $\mathbf{y}_{i_{t|k}}$, ($k = 1, ..., K$ $t = 1, ..., T_k$ $i = 1, ..., n$) which is equal to

$$\hat{\mathbf{u}}_{t|k} = E(\mathbf{u}_{t|k}|\mathbf{y}_{i_{t|k}}) = \mathbf{V}_k'(\mathbf{V}_k\mathbf{V}_k' + \mathbf{\Sigma}_{t|k})^{-1}(\mathbf{y}_{i_{t|k}} - \boldsymbol{\mu}_k). \tag{5.33}$$

It can be noticed that this estimator closely resembles the Thomson estimator (1951) of factor scores.

Whether $\mathbf{u}_{t|k}$ is random or nonrandom, differentiating $Q(\boldsymbol{\phi}, \cdot)$ with respect to $\boldsymbol{\mu}_k$ and equating the derivates to zero, we obtain the following updated estimate for $\boldsymbol{\mu}_k$:

$$\frac{\partial}{\partial \boldsymbol{\mu}_k}\left[\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T_k} w_{ik}w_{it|k}(\mathbf{y}_i - \boldsymbol{\mu}_k - \mathbf{V}_k\mathbf{u}_{t|k})'\mathbf{\Sigma}_{t|k}^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_k - \mathbf{V}_k\mathbf{u}_{t|k})\right] = 0,$$

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^{n} w_{ik}w_{it|k}(\mathbf{y}_i - \mathbf{V}_k\mathbf{u}_{t|k})}{\sum_{i=1}^{n} w_{ik}w_{it|k}}. \tag{5.34}$$

The double Gaussian hierarchical mixture model can be estimated through the upward-downward algorithm as follows:

- STEP E:

  Upward

  Updating of $\mathbf{W}_k = \{w_{it|k}\}$ and $\mathbf{W} = \{w_{ik}\}$ $(t = 1, ..., T_k,\ k = 1, ..., K)$;

  Downward

  Computation of $w_{ik}w_{it|k} = Pr(z_{ik} = 1, z_{it|k} = 1|\mathbf{y}; \boldsymbol{\phi})$.

- STEP M:

  Updating of $\pi = \{\pi_k\}$, $\pi_k = \{\pi_{t|k}\}$, $\mathbf{\Sigma}_k = \{\mathbf{\Sigma}_{t|k}\}$, $\mathbf{V}_k = \{v_{jh}\}$, $\mathbf{U}_k = \{\mathbf{u}_{t|k}\}$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_{t|k}$ $(t = 1, ..., T_k,\ k = 1, ..., K)$.

As in a standard EM algorithm, an iteration increases the observed data likelihood function and the sequence converges to a local optimum. There are a variety of heuristic approaches for escaping a local maximum such as using several different random initial estimates.

Thus, through the reparameterization of the 1st component-specific mean vector (5.27), we are able to cluster both units and variables, where conditionally to each unit cluster a different partition of the variables is allowed. In fact, the i-th unit is allocated to cluster k corresponding the largest posterior joint probability, $w_{ik}w_{it|k}$ ($t = 1, ..., T_k$, $k = 1, ..., K$); while j-th variable is allocated to the cluster q_k through the matrix membership $\mathbf{V}_k$ ($q_k = 1, ..., Q_k$, $k = 1, ..., K$).

5.3.5 Hierarchical VS standard mixture model

A drawback of this approach is that identifiability problems may arise. Willse and Boik (1999) discuss this issue underlining that, under the imposition of appropriate constraints, the two-level mixture model is identifiable. Hastie and Tibshirani (1996) use a two-level mixture model for discriminant analysis. However, their approach has not any identifiability problem because membership function is known. An important application field where constraints may be imposed, making the two-level mixture models identifiable, is illustrated by Di Zio et al. (2005).

In this Section, we will show that without additional constraints the hierarchical mixture model reduces to a standard mixture model (with a number of components equal to $\sum_{k=1}^{K} T_k = T$).

If we define $\pi_{tk} = \pi_{t|k}\pi_k$, we have

$$f(\mathbf{y}_i|\boldsymbol{\phi}) = \sum_{tk=1}^{T} \pi_{tk} f(\mathbf{y}_i|\boldsymbol{\theta}_{tk}). \qquad (5.35)$$

Hoveover, although the π_{tk}'s are sufficient to specify the mixture density $f(\mathbf{y}_i|\boldsymbol{\phi})$ according to (5.35), the individual component densities $f(\mathbf{y}_i|\boldsymbol{\theta}_{tk})$

cannot be determined without information about the (2nd level) cluster a (1st level) component belongs to.

To partially solve the ambiguity in assigning (1st level) components to (2nd level) clusters, we could choose the combination maximizing the log-likelihood function.

We briefly show an example of this concept, reminding that identifiability is clearly not attainable without further information. If we have some prior information on the number of (2nd level) clusters, we could pose constraints on cluster-specific means and this would greatly help model identifiability.

Let us fit a standard 4 component mixture model to the simulated data set displayed in Figure 5.1. It can be easily observed that the four clusters can be grouped in two macro clusters; each couple shows very similar features and its elements are very close to each other, while the two couples are quite far away. The estimated centroids of the 4 component solution are indicated in Figure 5.2 by a "+" sign. Then, we fit the hierarchical mixture model to the same data set and obtain the best clustering results (according to the

Figure 5.1: Simulated data set

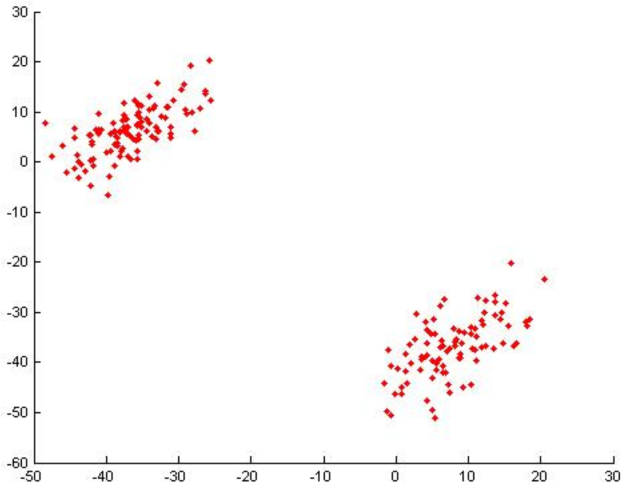
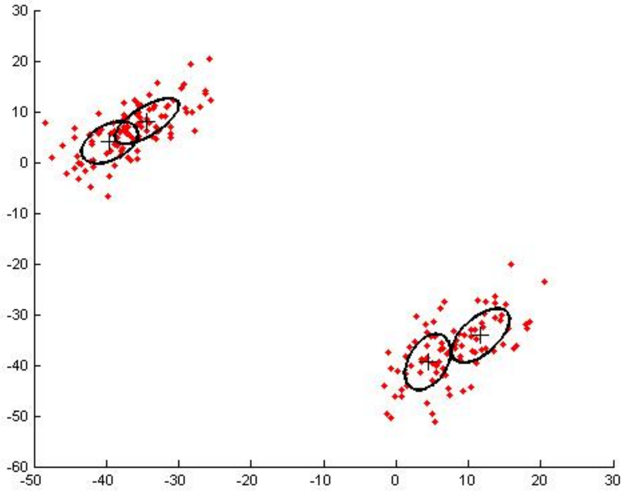


Figure 5.2: The centroids of four estimated clusters



BIC criterion) choosing two 2nd level clusters and two (1st level) components within each (2nd level) cluster. The first estimated (2nd level) cluster is exactly given by the first and second cluster of the standard solution while the other (2nd level) cluster is defined by grouping the third and fourth cluster. To get further insight, we considered all the alternatives obtained by joining the first and the second cluster (the third and the fourth), the first and the third (the second and the fourth) and finally, the first and the fourth (the second and the third). For each of these combinations, we fitted a standard mixture model with 2 components choosing the combination with the largest log-likelihood function value. The values of log-likelihood functions are -69.527, -73.763 and -149.330 respectively; therefore the best solution is given by the first combination with a log-likelihood value equal to -69.527. As outlined before, we obtained the same results fitting a standard mixture model with 4 components, by summing (1st level) components of the hierarchical mixture model.

5.3.6 Identifiability of $\mathbf{V}_k$

As mentioned before, the reparameterization of the mean vectors displayed in (5.27) is a particular case of the one proposed by Rocci and Vichi (2002). The main difference between the two reparameterizations is that Rocci and Vichi (2002) assume the matrix $\mathbf{V}_k$ is a factor loading matrix, and therefore the resulting model is not strictly identifiable since $\mathbf{V}_k$ can be rotated without affecting the results, provided that the latent vectors $\mathbf{u}_{t|k}$ are counter-rotated. On the other hand, we assume that the matrix $\mathbf{V}_k$ is binary and row stochastic and this help making the model identifiable. In fact, whether the vector $\mathbf{u}_{t|k}$ is a random vector or a vector of fixed effects, the membership matrix $\mathbf{V}_k$ is strictly identified. Let $\mathbf{V}_k$ be a $(J \times Q_k)$ binary matrix such as

$$v_{jq_k} = \begin{cases} 1 & \text{if the } j\text{-th variable belongs to the } q_k\text{-th cluster} \\ 0 & \text{otherwise} \end{cases} \tag{5.36}$$

with $\sum_{q_k=1}^{Q_k} v_{jq_k} = 1$ $(k = 1, ..., K)$. Let $\mathbf{P} = \{p_{q_k q_k}\}$ be a $(Q_k \times Q_k)$ orthogonal matrix (i.e., $\mathbf{P}' = \mathbf{P}^{-1}$) and $\mathbf{V}_k^* = \mathbf{V}_k\mathbf{P}$ a $(J \times Q_k)$ matrix such that

$$v_{jq_k}^* = \begin{cases} p_{q_k q_k} & \text{if } v_{jq_k} = 1 \\ 0 & \text{if } v_{jq_k} = 0 \end{cases}$$

with $\sum_{q_k=1}^{Q_k} v_{jq_k}^* = p_{q_k q_k}$ $(k = 1, ..., K)$.

In order for $\mathbf{V}_k^*$ to represent a membership matrix $(k = 1, ..., K)$, we must have

$$\sum_{q_k=1}^{Q_k} v_{jq_k}^* = \sum_{q_k \in \{q_k : v_{jq_k}=0\}} 0 + \sum_{q_k \in \{q_k : v_{jq_k}=1\}} p_{q_k q_k} = \sum_{q_k \in \{q_k : v_{jq_k}=1\}} p_{q_k q_k} = 1$$

for $j = 1, ..., J$ and $k = 1, ..., K$. Then, the orthogonal matrix $\mathbf{P}$ such that the binary matrix $\mathbf{V}_k^*$ can be written as $\mathbf{V}_k^* = \mathbf{V}_k\mathbf{P}$ with the constraint $\sum_{q_k=1}^{Q_k} v_{jq_k}^* = 1$ is the identity matrix $\mathbf{I}_{Q_k}$. In other words, $\mathbf{V}_k^* = \mathbf{V}_k$, i.e. $\mathbf{V}_k$ is unique.

5.3.7 Conclusions

We have proposed an approach obtained by adapting the multilevel latent class model proposed by Vermunt (2003) to two-way continuous data; observations are clustered into a particular (1st level) latent component within a certain (2nd level) cluster. In order to cluster variables we have introduced a binary and row stochastic matrix of variable cluster membership (as in Vichi, 2000). We have discussed a potential reparameterization of (1st level) mean vectors of component-specific densities according to the work of Rocci and Vichi (2002). We have tested the proposed model using a simulation study and obtained encouraging results (see Part II of dissertation).

It can be observed that, thanks to the hierarchical structure, we learn the ground-truth data clusters by distinguishing the number of components (1st level components) by the number of clusters (2nd level components) which is a standard problem in Gaussian mixture model (see Shental, Bar-Hillel, Hertz and Weinshall, 2003; Zhao and Miller, 2005).

In fact, observations are assumed to be independent given (1st level) component membership: the hierarchical structure accounts for some dependence between units showing similar behaviour, i.e. belonging to the same cluster. Within each cluster, we model departures from standard Gaussian assumptions using a number of components representing heterogeneity with respect to the Gaussian case. Thus, heterogeneity may affect (2nd level) cluster-specific distributions in a number of ways according to the shape of the *real* distribution and the number of (1st level) components which is needed to recover it.

Moreover, thanks to the particular specification of the hierarchical model displayed in (5.27), we can cluster both units and variables identifying *blocks*, i.e., sub-matrices of the observed data matrix, where units and variables specify a *2nd level cluster* and a *variable cluster* (specific for each 2nd level cluster).

As we have noticed in Section 5.3.5, the hierarchical mixture model is not globally identifiable unless, for example, we are able to fix the number of (2nd level) clusters, K; in this case, we use the remaining degrees of freedom to identify (1st level) components within each (2nd level) cluster. This discussion shows that identifiability is not more problematic for the hierarchical mixture model than for the standard mixture model.

Finally, the model suggests the use of BIC or AIC as penalizing function for choosing the number of 1st as well as 2nd level components and of variable clusters.

Part II

APPLICATIONS

Chapter 6

Simulation Studies

To test the performance of the double clustering based on the proposals described in the previous Chapter, experiments have been carried out with both simulated and real data sets. In this Chapter, we will deal with the simulation studies, where the performance of model-based double K-means and hierarchical mixture model have been evaluated in terms of recovery of true partitions. In the next Sections, we see the results in greater detail.

6.1 Performance in simulation study of double K-means by using a ML approach

The performance of the (model-based) double K-means has been tested in a simulation study by employing $b = 1, ..., B = 100$ runs. Model parameters have been estimated by using the algorithm2rid, described in Section 5.1.3, characterized by a CPU time which is substantially lower than the other proposed algorithms. In particular, the performance of the algorithm has been evaluated by using the following measures:

1. Mrand($\mathbf{U}$, $\mathbf{U}_b$). Modified Rand Index (Hubert and Arabie, 1985) between the true matrix $\mathbf{U}$ and the estimated matrix $\mathbf{U}_b$, for each run $b = 1, ..., B = 100$;
2. Mrand($\mathbf{V}$, $\mathbf{V}_b$). Modified Rand Index between the true matrix $\mathbf{V}$ and the estimated matrix $\mathbf{V}_b$, for each run $b = 1, ..., B = 100$;
3. number of times the fitted partition of units is equal to the true partition, i.e., Mrand($\mathbf{U}$, $\mathbf{U}_b$)=1, for each run $b = 1, ..., B = 100$;
4. number of times the fitted partition of variables is equal to the true partition, i.e., Mrand($\mathbf{V}$, $\mathbf{V}_b$)=1, for each run $b = 1, ..., B = 100$.

As observed in Section 5.1.3, the algorithm begins with random starting values $\mathbf{U}$ and $\mathbf{V}$; in the current context, we have observed that few random starts are generally enough to find an optimal solution. However, to be sure that the optimal solution is detected, the number of random starts has been fixed equal to ten.

In each experiment, B data sets have been generated according to model (5.1) in a $J = 80$ dimensional space with a number of units equal to $n = 2000$; each row, $\mathbf{x}_i$, is supposed to be drawn from a J-variate Gaussian distribution with mean vector $\mathbf{V}\bar{\mathbf{X}}'\mathbf{u}_i$ and covariance matrix $\mathbf{\Sigma}$. Then, blocks are randomly placed within the data matrix.

Three error levels (Low, Medium, High) have been considered in order to have different levels of homogeneity within blocks. The error levels have been used to multiply the covariance matrix, $\mathbf{\Sigma}$), by 5,10, 50 respectively. In Figures 6.1 and 6.2 (a, b, c), we show the data set types with Low, Medium and High; if error level is low, the structure of blocks is well visible. On the contrary, if the error level is high the structure of blocks is lost. Then, rows and columns have been randomly permuted and the designed algorithm

Number of generated data sets	B=100
Number of units	n=2000
Number of variables	J=80
Number of units of clusters	K=3, 6
Number of variables of clusters	Q=2, 4
Number of random starts	10
Error level (trace($\Sigma = \sigma^2\boldsymbol{\Omega}$))	*Low* $\sigma^2 = 5$, *Medium* $\sigma^2 = 10$, *High* $\sigma^2 = 50$

Table 6.1: Simulation setting

has been applied to recover blocks and specifically partitions of units and variables generating the data (Figure 6.1 and 6.2; a2, b2, c2).

We have considered two different situations:

1. the data matrix is formed by 6 blocks defining a partition of units with K=3 clusters and a partition of variables with Q=2 clusters;

2. the data matrix is formed by 24 blocks defining a partition of units with K=6 clusters and a partition of variables with Q=4 clusters.

The design of the simulation experiments is shown in Table 6.1, while Table 6.2 and Table 6.3 display the simulation results with K=3, Q=2 and K=6, Q=4 respectively. It can be observed that the algorithm performs well in recovering the true partitions of units and variables under all levels of error.

In general, the algorithm performs better in recovering the partition of variables rather than that of units (see columns 2-5, Tables 6.2-6.3); by increasing the error level, the average Mrand index for both units and variables partitions slowly decreases.

Finally, as far as the CPU time in both experiments is concerned, we can observe that the algorithm converges quite fastly (2 or 3 iterations) even if it

Error Level	Average Mrand for units partitions	Average Mrand for variables partitions	% of unit partitions= to the true partitions	% of variable partitions= to the true partitions	Average number of iterations
Low	1	1	1	99	2.09
Medium	0.98	0.99	92	100	2.25
High	0.95	0.98	71	100	3.56

Table 6.2: Simulation results: K=3, Q=2

begins with random starting values for **U** and **V**. In order to further speed up its convergence, those values could be chosen in a rationale way (e.g., using some clustering method).

Figure 6.1: (2000×80) data matrices subdivided into $K = 3$ unit clusters and $Q = 2$. Three levels of error a), b) c) are considered. The algorithm2rid has been applied by permuting of rows and columns of the data matrices

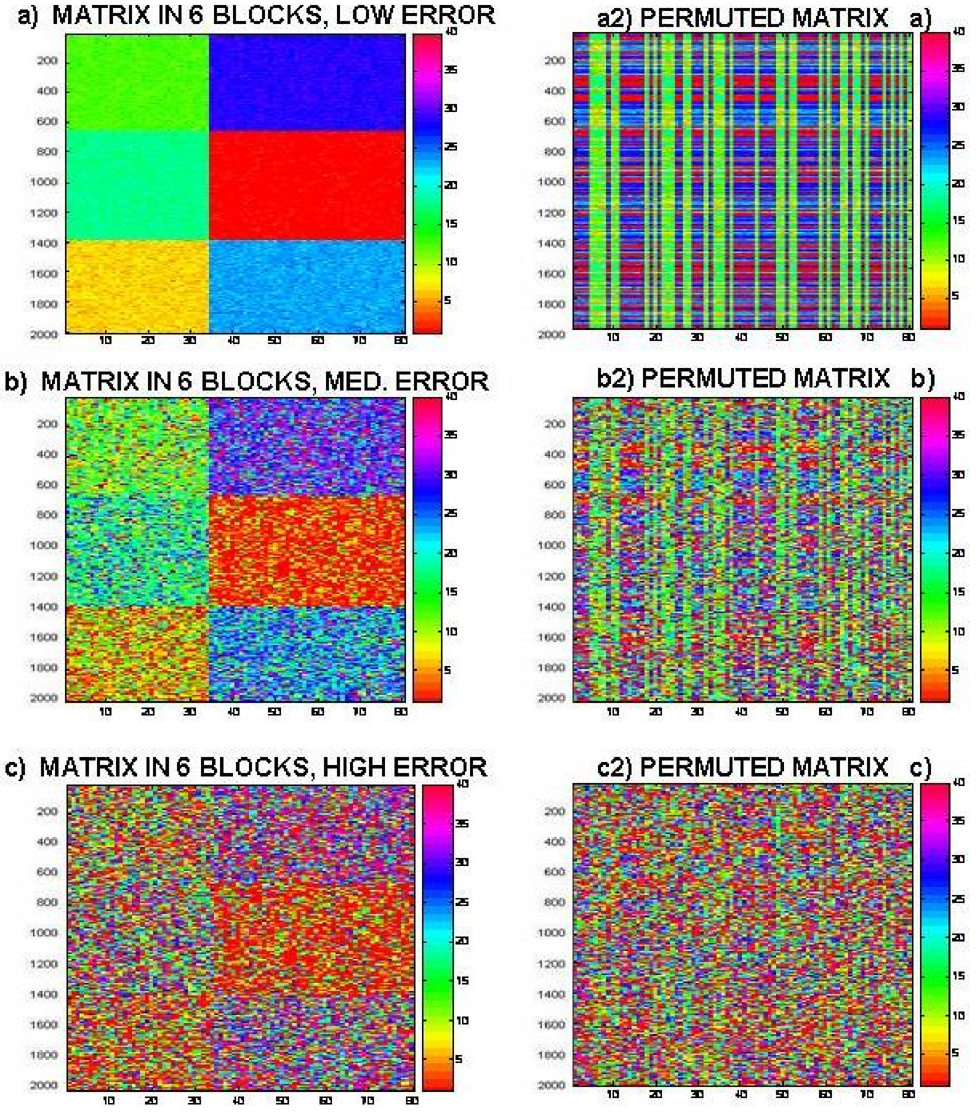
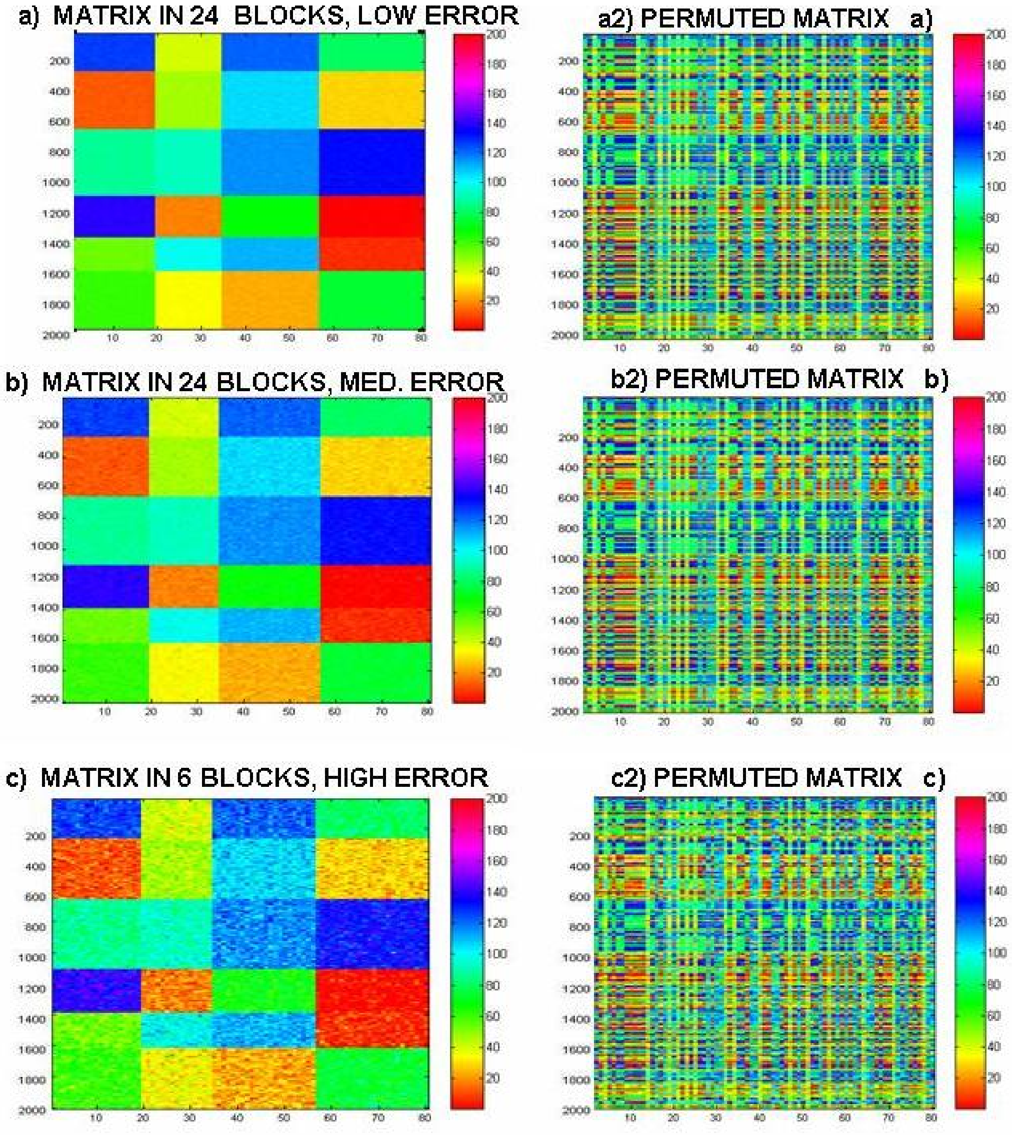


Figure 6.2: (2000 × 80) data matrices subdivided into $K = 6$ unit clusters and $Q = 4$. Three levels of error a), b) c) are considered. The algorithm2rid has been applied by permuting of rows and columns of the data matrices



Error Level	Average Mrand for units partitions	Average Mrand for variables partitions	% of unit partitions= to the true partitions	% of variable partitions= to the true partitions	Average number of iterations
Low	1	0.98	84	96	2.19
Medium	0.96	0.99	76	97	2.18
High	0.93	0.97	20	70	3.27

Table 6.3: Simulation results: K=6, Q=4

6.2 Performance in simulation study of the hierarchical mixture model

The performance of the hierarchical mixture (H-M) model has been tested using two simulation studies. The first experiment was conducted to evaluate the performance of the hierarchical mixture model shown in Equations (5.12) to (5.14) in terms of recovery of the true row partition. The second experiment was carried out to evaluate the performance of the hierarchical model with the reparameterization of mean vectors given in (5.27) in terms of recovering both row and column partitions. In particular, for the first experiment we have evaluated the performance of the upward-downward algorithm through the following measures:

1. Mrand($\mathbf{Z}$, $\mathbf{W}^*$). Modified Rand Index between $\mathbf{Z} = \{z_{ik}\}$, the true matrix of (2nd level) cluster membership and the estimated matrix

$\mathbf{W}^* = \{w^*_{ik}\}$, where each element w^*_{ik} is defined as:

$$w^*_{ik} = \begin{cases} 1 & \text{if } k = \text{argmax}_{k=1,\dots,K} w_{ik}, \\ 0 & \text{otherwise} \end{cases}; \tag{6.1}$$

2. Mrand($\mathbf{Z}_k$, $\mathbf{W}^*{}_k$). Modified Rand Index between $\mathbf{Z}_k = \{z_{it|k}\}$, the true matrix of (1st level) component membership and the estimated matrix $\mathbf{W}^*{}_k = \{w^*_{it|k}\}$ $(k = 1, ..., K)$, where each element $w^*_{it|k}$ is defined as:

$$w^*_{it|k} = \begin{cases} 1 & \text{if } t = \text{argmax}_{t=1,\dots,T_k} w_{it|k}, \\ 0 & \text{otherwise} \end{cases}. \tag{6.2}$$

For the second experiment, we use as additional measure of goodness of fit the value of Mrand($\mathbf{V}^*_k$, $\mathbf{V}_k$), the Modified Rand Index between the true matrix $\mathbf{V}_k$ and the estimated matrix $\mathbf{V}^*_k$ $(k = 1, ..., K)$.

In the first simulation setting, the algorithm starts with random matrices $\mathbf{Z}$ and $\mathbf{Z}_k$ $(k = 1, ..., K)$; random starts in the second scenario include also the random matrix $\mathbf{V}_k$. For both experiments, the number of random starts is fixed equal to ten, because we have observed that a higher number of random starts does not increase the chance to find the optimal solution.

The first experiment design is summarized in Table 6.4: 100 data sets have been generated according to Equations (5.12) to (5.14) in a J=30 dimensional space with a number of units equal to n=2000, $K = 2$ (2nd level) clusters and $T_1 = T_2$=2 (1st level) component for each (2nd level) cluster. Table 6.5 displays the design of the second experiment, where 100 data sets have been generated according to Equations (5.12) to (5.14) with the mean reparameterization given in (5.27) and J=30, $n = 2000$, $K = 2$, T_1=2, T_2=2 and $Q_1 = Q_2$=2.

For each experiment and each data set, partitions (respectively blocks, i.e. sub-matrices of the observed data matrix) have been randomly placed by permuting rows and columns (an example is given in Figures 6.3 and 6.4). Three error levels (Low, Medium and High) have been considered in order to work with varying levels of homogeneity within partitions (blocks). Those error levels have been fixed multiplying the covariance matrix by 5, 10, 50 respectively. As can be easily observed in Figures 6.3 and 6.4 a), if error level is low partitions (blocks) are well distinguished; on the other hand, in Figures 6.3 and 6.4 c) error levels are high and partitions (blocks) can be hardly recognized. In Figure 6.3 and Figure 6.4 a2), b2), c2) rows and columns have been randomly permuted to mask the clustering structure. Then, the proposed model has been estimated to recover the original partitions (blocks) and specifically the "true" partition of units and/or variables. Tables 6.6 and 6.7 display the simulation results.

As can be observed from both studies, the proposed model performs well in recovering the true partition of units and/or variables under all error levels. In general, when the error level increases, the average Mrand for objects and/or variables partitions slowly decrease (see columns 2-3 in Table 6.6 and columns 2 to 5 in Table 6.7).

For both experiments, the algorithm begins by using as starting points the solution of the algorithm2rid for double K-means. This procedure improves the performance of the upward-downward algorithm considerably requiring a small number of iterations for both the 1st and the 2nd level as shown in Table 6.6 and 6.7.

Figure 6.3: (2000 × 30) data matrices with K=2, $T_1 = 2$, $T_2 = 2$. Three error levels a), b) and c) are considered. Entries are randomly permutated

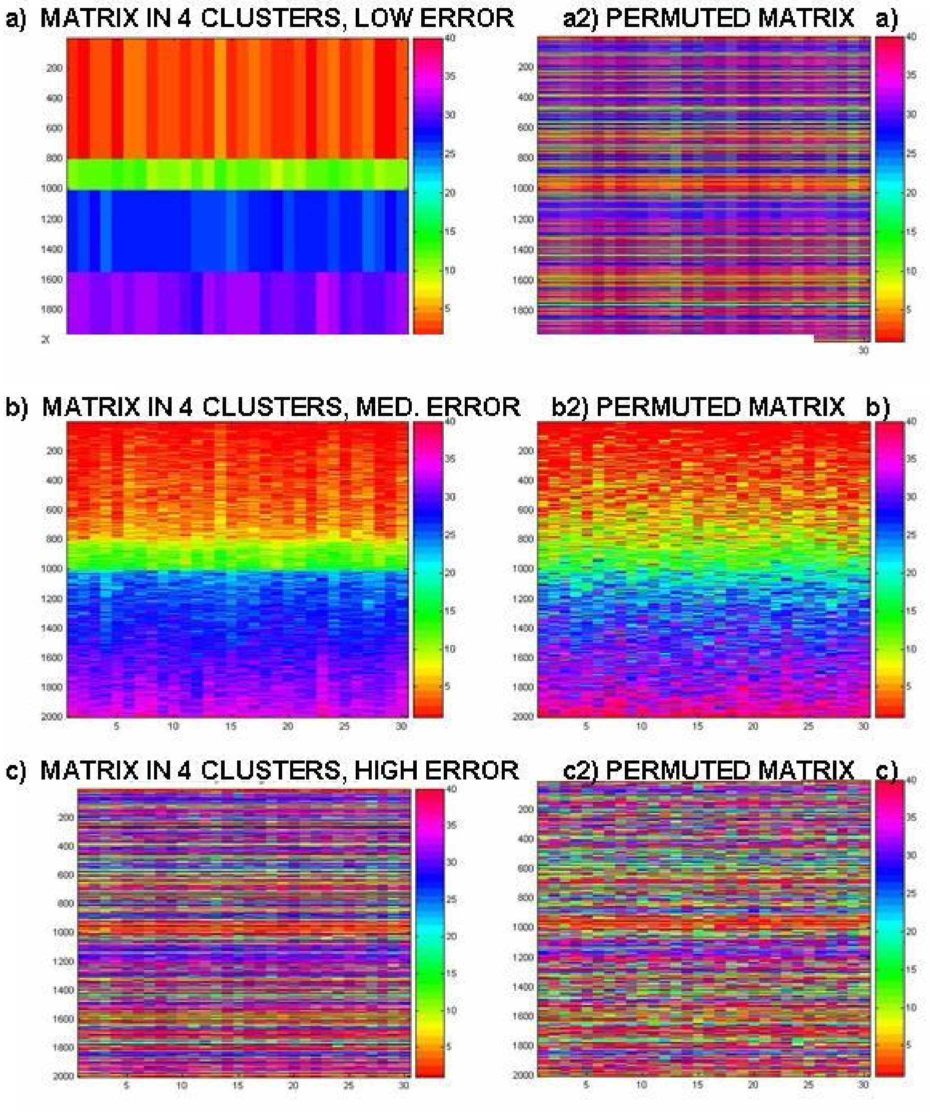
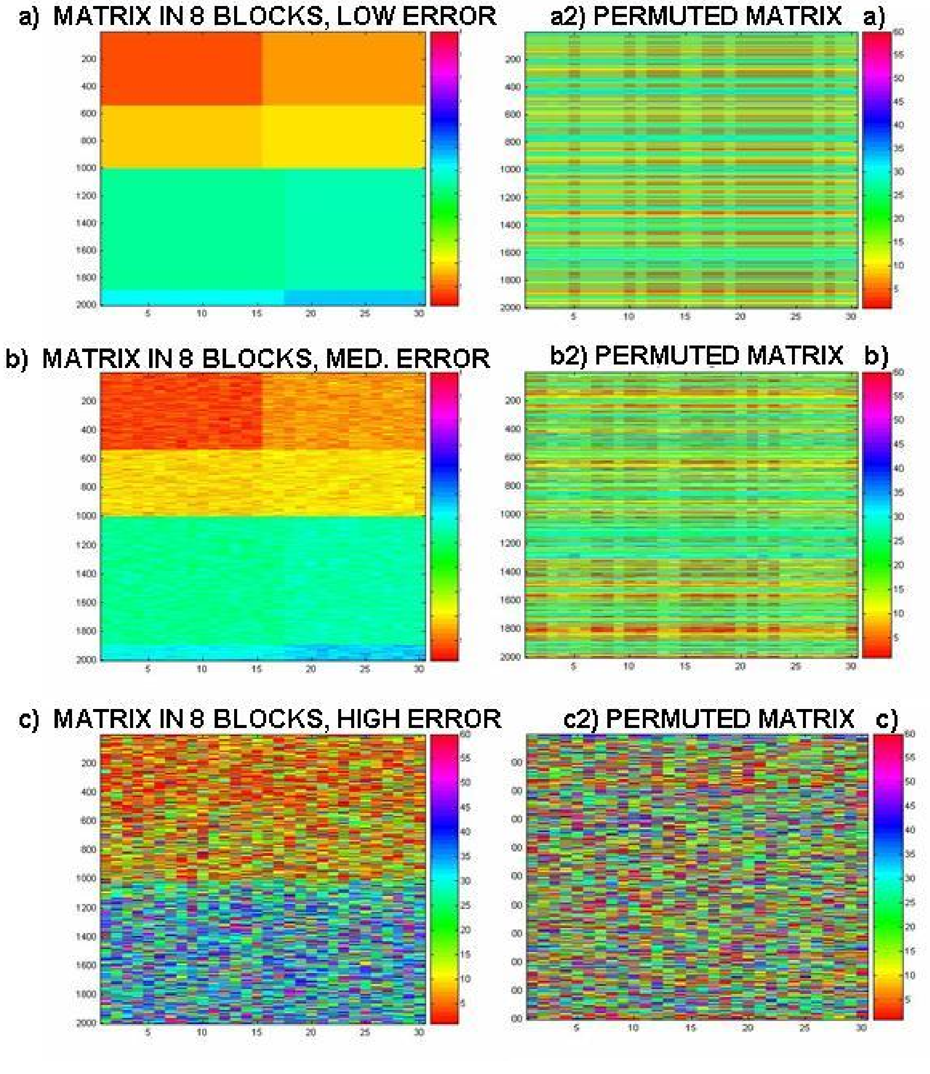


Figure 6.4: (2000×30) data matrices with K=2, $T_1 = 2$, $T_2 = 2$ and $Q_1 = Q_2 = 2$. Three error levels a), b) and c) are considered. Entries are randomly permutated



Number of generated data sets	B =100
Number of units	n=2000
Number of variables	J=30
Number of 2nd level clusters	K=2
Number of 1st level sub-clusters	$T_1 = 2$, $T_2 = 2$
Number of random starts	10
Error level (trace($\Sigma = \sigma^2\mathbf{\Omega}$))	*Low* $\sigma^2 = 5$, *Medium* $\sigma^2 = 10$, *High* $\sigma^2 = 50$

Table 6.4: Simulation design for the first experiment

Number of generated data sets	B =100
Number of units	n=2000
Number of variables	J=30
Number of 2nd level clusters	K=2
Number of 1st level sub-clusters	$T_1 = 2$, $T_2 = 2$
Number of variables clusters	$Q_1 = Q_2 = 2$
Number of random starts	10
Error level (trace($\Sigma = \sigma^2\mathbf{\Omega}$))	*Low* $\sigma^2 = 5$, *Medium* $\sigma^2 = 10$, *High* $\sigma^2 = 50$

Table 6.5: Simulation design for the second experiment

<table>
<tr><th rowspan="2">Error level</th><th rowspan="2">Average Mrand for 2nd level units partitions</th><th rowspan="2" colspan="2">Average Mrand for 1st level units partitions</th><th colspan="2">Average number of iterations for 2nd level</th><th rowspan="2">Average number of iterations for 1st level</th></tr>
<tr><th>T_1</th><th>T_2</th></tr>
<tr><td>Low</td><td>1.00</td><td>1.00</td><td>1.00</td><td colspan="2">2.00</td><td>20.65</td></tr>
<tr><td>Medium</td><td>1.00</td><td>0.98</td><td>1.00</td><td colspan="2">2.19</td><td>22.10</td></tr>
<tr><td>High</td><td>1.00</td><td>0.99</td><td>0.80</td><td colspan="2">2.95</td><td>27.00</td></tr>
</table>

Table 6.6: Simulation results for the first experiment

Error level	Average Mrand for 2nd level units partitions	Average Mrand for 1st level units partitions		Average Mrand for variables partitions		Average number of iterations for 2nd level	Average number of iterations for 1st level
		T_1	T_2	T_1	T_2		
Low	1.00	1.00	1.00	1.00	1.00	2.10	22.30
Medium	1.00	1.00	0.87	0.99	0.98	2.50	28.50
High	0.99	0.89	0.80	0.88	0.74	3.05	30.95

Table 6.7: Simulation results for the second experiment

Chapter 7

Microarray case studies

As far as real data sets are concerned, we have focused on one of the most novel fields in the bioinformatic area, namely gene expression data. Despite information obtained from ongoing research on sequencing the human genome (structural genomics), we still lack full understanding of how our genes are turned on or off to maintain a healthy body (functional genomics). Many diseases, including cancer genetic diseases and other infectious diseases, are direct consequences of mis-expression of the genes. To examine how regulatory proteins assemble a gene and regulate its expression, we need to know the expression levels of thousands of genes in the same conditions. These data are available through microarray experiments that are performed over a set of conditions. The dimension and complexity of raw gene expression data obtained by oligonucleotide or spotted microarrays, create challenging data analysis and management problems ranging from the analysis of images produced by microarray experiments to biological interpretation of results. Therefore, statistical and computational approaches are beginning to assume a substantial position within the molecular biology area.

The use of clustering techniques is essential in the data mining process to reveal natural structures and identify interesting patterns in the gene expression data.

In this empirical context, the aim could be the clustering of both genes and tissue samples; clustering based on either genes or tissue samples may in fact be unable to give insightful results. The need to find a subset of genes and tissue samples defining a homogeneous block had led to the application of double clustering techniques on gene expression data (see Chapter 4).

The standard methods of double clustering do not utilize the inherent statistical structure of data; these methods are often pulled within the category of heuristic methods. We have proposed a method for double clustering using a model-based approach, taking advantages of the probabilistic framework previously discussed.

In the following Section, a description of the microarray technology is introduced. A short description of analyzed data sets will be provided in Section 7.2 and 7.4. The results obtained from analyzing the gene expression data sets by using the double K-means and the double hierarchical mixture model are included in Sections 7.3.1 and 7.3.2 for Bittner et al. (2000) data and, respectively, 7.5.1 and 7.5.2 for Golub et al. (1999) data.

7.1 The microarray technology

7.1.1 Genetic Overview

The *cell* is the structural and functional unit of a living organism. The *DNA* (Deoxyribonucleic acid), usually in the form of a double helix, is the most important of all cell molecules since it contains genetic instructions (*genome*) monitoring the biological development of cellular forms of life, as well as of many viruses. The genome represents the whole hereditary information of an organism and includes both the genes and the non-coding sequences. The term was coined in 1920 by Hans Winkler, Professor of Botany at the University of Hamburg, Germany, as a portmanteau of the words gene and chromosome. More precisely, the genome of an organism is a complete DNA sequence of one set of *chromosomes* with specific number and form for each specie. The individual genome is constituted by 46 chromosomes: 22 pairs of autosomes and two sex chromosomes. Each chromosome contains some *genes*, which identify functional regions of DNA; their function is to encode the necessary instructions in order to produce the proteins. About 98.5% of the human genome has been designated as *junk DNA* which is the portion of the DNA sequence for which no function has yet been identified. Human DNA contains more than 30.000 different genes. DNA consists of two strands, being complementary to each other. The strands are made up of four basic nucleotides (or bases): adenine (A); thymidine (T); cytosine (C) and guanine (G). The A (T) base of DNA strand only bonds with the T (A) base from the complementary strand (cDNA), likewise for C and G. The complementary principle forms the foundation of DNA microarrays.

To build proteins, the genetic information in DNA is transcribed to an intermediate product in the nucleus of cells, the mRNA (messenger RNA). Genetic instructions are then carried by mRNA for translation into proteins. Gene expression is measured as the amount of mRNA for a particular gene in the analyzed cell. In fact, genes are expressed in different ways in different body. To give an example, let us consider genes expressed in pancreas. Some genes are expressed in the pancreas and in many or all other tissues in the body (these are called housekeeping genes). Many genes are not expressed in the pancreas, but are expressed in other tissues. A few genes are expressed in the pancreas but not in other tissues, and in this case mRNA (and protein) encoded by these genes will only be present in the pancreas cells (that is, over expressed) but not in other tissues (that is, under expressed). In the case of pancreas cancer, most of the genes that are present in healthy pancreas will also be expressed in the cells of pancreas cancer. However, certain genes will be over-expressed or under-expressed in the cancer cells compared to normal pancreas cells. This modified gene expression contributes to uncontrolled growth and spread of the tumor.

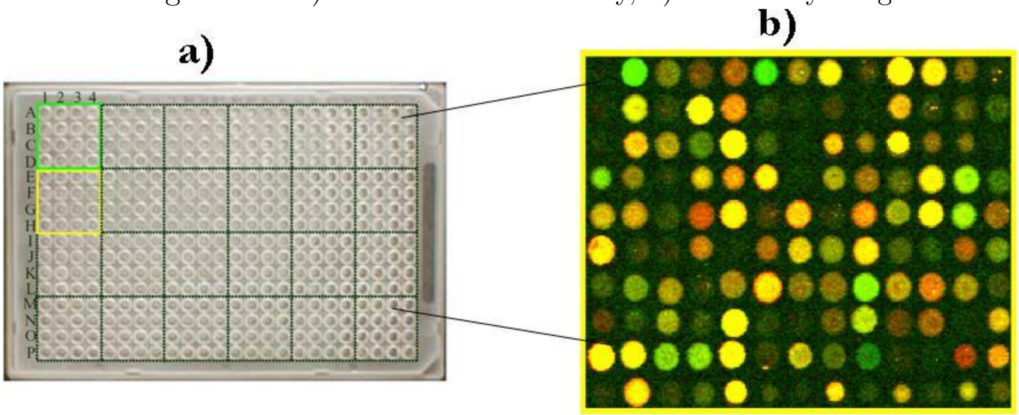
7.1.2 Types of microarrays

A microarray (also known as gene chip, DNA chip, or biochip) is typically a glass slide, where DNA molecules are attached at fixed locations (spots or features) forming an array for monitoring expression levels corresponding to thousands of genes simultaneously. Tens of thousands spots may be present over an array, each containing a huge number of identical DNA molecules (or fragments of identical molecules), of length from twenty to hundreds nucleotides. The spots on a microarray are either printed by a robot, or synthesized by photo-lithography (similar to computer chip productions) or by ink-jet printing. Microarrays containing the about 6000 genes of the yeast genome have been available since 1997 (Science, 1997). Latest generation of commercial microarrays represent the entire human genome, more than 30.000 genes, on two microarrays. Microarrays can be fabricated using a variety of technologies, including printing with fine-pointed pins onto glass slides, photolithography using pre-made masks, photolithography using dynamic micromirror devices, ink-jet printing, or electrochemistry on microelectrode arrays. In the following Sections, we will briefly describe the most usual microarray types.

Spotted microarrays

In spotted microarrays (also known as two-channel microarrays), the probes are oligonucleotides, cDNA or small fragments of PCR (Polymerase Chain Reaction) products corresponding to mRNAs. This type of array is typically hybridized with cDNA from two samples that have to be compared (e.g. patient and control) and are labeled with two different fluorofores (Figure 7.1). The samples can be mixed and hybridized to one single microarray to be scanned, allowing the visualization of up-regulated and down-regulated genes in one go. The drawback of this approach is that the absolute levels of gene expression cannot be observed; however the cost for the experiment is reduced by half.

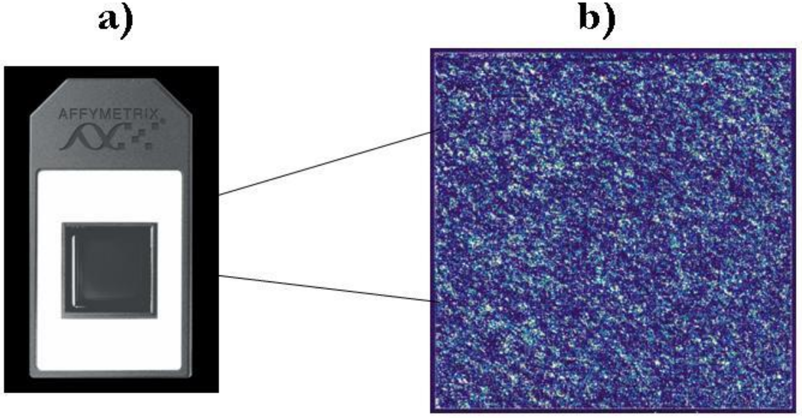
Figure 7.1: a) two-channel microarray, b) microarray image



Oligonucleotide microarrays

In oligonucleotide microarrays (or single-channel microarrays), the probes are designed to match parts of the sequence of known or predicted mRNAs. There are commercially available designs that cover complete genomes from companies such as GE Healthcare, Affymetrix (Figure 7.2), or Agilent. These microarrays give a measure of the (absolute value) gene expression and therefore the comparison of two conditions requires the use of two separate slides. Oligonucleotide Arrays can be either produced by piezoelectric deposition with full length oligonucleotides or by in-situ synthesis.

Figure 7.2: a) Affymetrix microarray, b) microarray image



Long Oligonucleotide Arrays are composed of 60-mers, and are produced by ink-jet printing on a silica substrate. Short Oligonucleotide Arrays are composed of 25-mer or 30-mer and are produced by photolithographic synthesis (Affymetrix) on a silica substrate or piezoelectric deposition (GE Healthcare) on an acrylamide matrix. More recently, Maskless Array Synthesis from NimbleGen Systems has combined flexibility with large numbers of probes. Arrays can contain up to 390000 spots, from a custom array design. New array formats are being developed to study specific pathways or disease states for a systems biology approach.

Genotyping microarrays

SNP microarrays are a particular type of DNA microarrays that are used to identify genetic modification in individuals and across populations. Short oligonucleotide arrays can be used to identify the single nucleotide polymorphisms (SNPs) that are thought to be responsible for genetic modification and to be the source of genetically caused diseases. Generally termed genotyping applications, DNA microarrays may be used in this fashion for forensic applications, rapidly discovering or measuring genetic predisposition to a given disease, or identifying DNA-based drug candidates.

These SNP microarrays are also being used to profile somatic mutations in cancer. Amplifications and deletions can also be detected using comparative genomic hybridization in conjunction with microarrays.

Resequencing arrays have also been developed to sequence portions of individual genome. These arrays may be used to evaluate germ-line mutations in individuals, or somatic mutations in cancer.

Genome tiling arrays include overlapping oligonucleotides designed to blanket an entire genomic region of interest. Many companies have successfully designed tiling arrays that cover whole human chromosomes.
In the next Section, we will describe how cDNA microarrays are made.

7.1.3 Description of the cDNA microarray experiment

In a generic cDNA microarray experiment, the basic idea is the comparison of gene expression levels between pairs of samples, such as a ill and a healthy tissue. This comparison is made through the following steps:

1. Extraction of mRNAs from test (e.g. tumor) and reference (e.g. healthy) samples;
2. Retro-transcription to cDNA (to improve the measurement of mRNA);
3. The two samples are labelled with two different fluorescent dyes of cDNA (e.g. red and green);

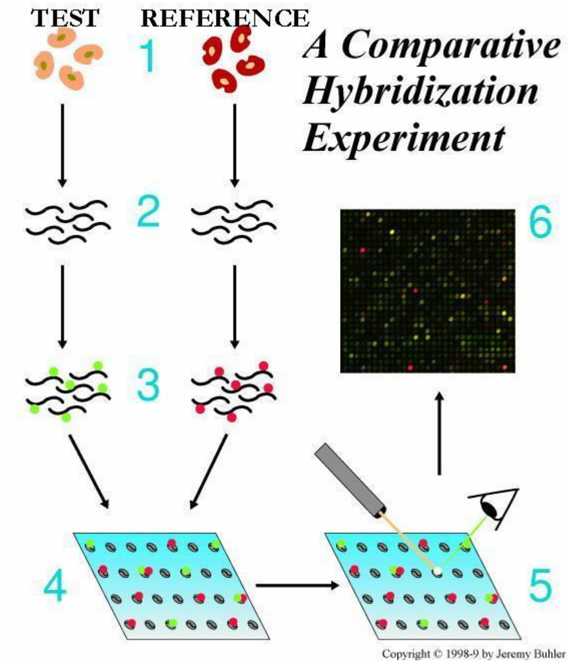
4. Mixing and hybridization of labelled cDNA molecules on a microarray. In such a hybridization, labelled cDNA molecules bind to their complementary sequences on the microarray;

5. After hybridization, the microarray slide is scanned and the resulting image is stored as a 16-bit TIFF image.

6. Location of the cDNAs on a microarray, or spots, are identified using a software for image analysis.

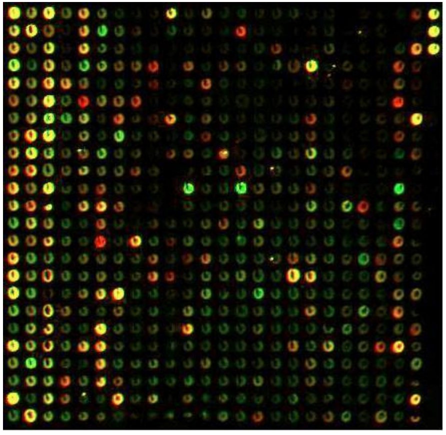
Such steps are shown in Figure 7.3.

Figure 7.3: A comparison of gene expression levels between test and reference sample


TEST
REFERENCE
A Comparative Hybridization Experiment
1
2
3
4
5
6
Copyright © 1998-9 by Jeremy Buhler

The final result of the experiment is shown in Figure 7.4. The red spot indicates an up-regulated gene, while a green spot indicates a down-regulated gene. Genes showing some equal expressions in test and reference samples are illustrated using yellow color.

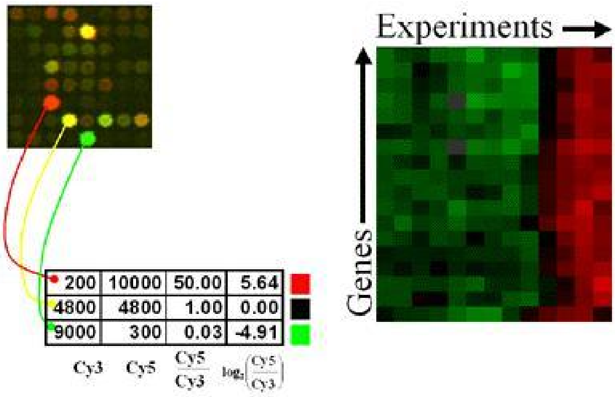
Figure 7.4: Microarray image



7.1.4 Converting the scanned image to the spotted image

The final step is converting the scanned image in a numeric table (Figure 7.5). The task of quantifying a scanned image is defined as follows.

Figure 7.5: Conversion of a microarray image in numeric form



Cy3	Cy5	$\frac{Cy5}{Cy3}$	$\log_2\left(\frac{Cy5}{Cy3}\right)$	
200	10000	50.00	5.64	🟥
4800	4800	1.00	0.00	⬛
9000	300	0.03	-4.91	🟩

Each spot is segmented, which means that a border between actual signal and background noise is determined. For each segmented spot, the average test intensities is computed and divided by the average reference intensities. The median of test and reference intensities is also used instead of the mean.

The ratios, indicate relative expressions of genes in the two samples. The outcome of a microarray experiment is typically a table of ratios measuring the relative expression levels of each gene on a microarray.

7.1.5 Preprocessing Microarray Data

Even if a microarray experiment is carefully designed, error sources causing variation in the analyzed levels are frequently observed. In order to reduce the effects of these error sources, microarray data analysis requires some preprocessing steps. Methods used in preprocessing may differ between cDNA microarray data and oligonucleotide microarray data. Some of the most common error-sources preventing the direct application of statistical techniques are:

- Variation of the amount of DNA in the microarray spots;
- Systematic variation in printing pin groups (print-tip bias);
- Different physical characteristics between the dyes used for labelling (dye bias);
- Unequal amount of mRNA in test and reference samples;
- Unequal background intensity of scanned microarrays;
- Impurities such as dust particles on microarrays;
- Variation between microarray slides;
- Variation in experimental conditions.

Preprocessing consists of three phases: *quality control*, *within-slide normalization*, and *multiple-slide normalization*. Quality control is usually done by excluding spots which are considered unreliable, as a result of low signal-to-noise ratio or small spot area. Within-slide normalization methods aim at balancing the test and reference intensities, while the main purpose of multiple-slide normalization is to ensure that data from different slides are comparable.

For a complete review of microarray technology and statistical issues, refer to Amaratunga and Cabrera (2004). Many microarray databases can be accessed via internet; for example, http://genome-www5.stanford.edu/ or http://proteogenomics.musc.edu/.

7.2 Description of the Bittner et al. data set (2000)

In this Section a short description of the analyzed benchmark data set on cutaneous melanoma will be provided. For more details please see Bittner et al. (2000). The data are available from the web site:
http://www.nhgri.nih.gov/DIR/Microarray/Melanoma_Supplement/index.html.

The original aim of this study was to determine whether or not molecular profiles generated by cDNA microarrays could be used to identify distinct subtypes of cutaneous melanoma, a malignant neoplasm of the skin. The data consists of 38 samples from tissue biopsies and tumor cell lines, with 31 cutaneous melanomas and 7 controls; samples come from male and female patients aged 29 to 75, with 3 patients of unknown age. The mRNA was extracted and Cy5-labelled cDNA was created for the 31 cutaneous melanoma and the 7 control samples; a single reference probe, labelled Cy3, was used for all the 38 samples. The Cy5 and Cy3-labelled cDNAs are mixed for each sample and hybridized to a separate melanoma microarray. The hybridized array was scanned using both red and green lasers, and the resulting image was analyzed. 3613 over 8150 cDNAs were identified as adequately measured, and gene expression ratios of Cy5/Cy3 were calculated. Ratios greater than 50 and lower than 0.02 were truncated to 50 and 0.02, respectively; often that ratios were transformed to a logarithm scale (base 2); and normalized by subtracting the median log-ratio for that experiment, so that the median log-ratio within an experiment was zero. No normalization was performed across experiments, since a single reference probe was used for all of them. Bittner et al. (2000) discuss the analysis of 31 samples, excluding the 7 control samples. Average linkage hierarchical clustering was carried out on these 31 samples by using one minus the Pearson correlation coefficient between log-ratios as a dissimilarity measure between two experiments. In this way, they obtained two clusters of 12 and 19 samples, which have been validated by multidimensional scaling (MDS) and through CAST, a non-hierarchical clustering algorithm (Ben-Dor, Shamir and Yakhini, 1999). Both MDS and CAST identified the same major cluster of 19 samples found by the average linkage hierarchical clustering.

7.3 Results

Here, we focus on double clustering of genes and tissue samples, by using the models described in Sections 5.1 and 5.3.

We denote the observed gene expression levels by $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n$, where $\mathbf{y}_i$ is a 31-dimensional vector representing the expression level of the i-th gene on 31 cutaneous melanoma samples. Thus, we have $J = 31$ and $n = 3613$.

After centering and rescaling columns to unit variance to be coherent with the analysis of Bittner et al. (2000) [1], we applied both proposed models with the aim of defining significant blocks formed by gene clusters characterizing subtypes of cutaneous melanomas.

7.3.1 Double K-means

We have fitted the double K-means using the ML approach described in Section 5.1 for different values of number of clusters for genes and tissues. For each pair, we run the algorithm2rid several times to avoid local minima, choosing the best solution through BIC and AIC criteria; in particular, the combination providing the lowest values for those criteria. Table 7.1 summarizes the most significant results. As it can be noticed, the best solution corresponds to 8 blocks in data matrix, e.g. 4 genes clusters (with cardinality 83-707-1612-1211) and 2 tissue samples clusters (21-10) (see line 3 in Table 7.1); these are displayed in Figure 7.6. However, only 2 blocks (as showed in Figure 7.7) seem to be biologically meaningful; the others have block mean close to zero and this suggests that the gene expression levels in these tissue samples are equal to the gene expression levels in the reference. In particular, the 2 meaningful blocks showing block means equal to -2.54 and -1.84, are formed by 83 down-regulated genes classified in two clusters of 21 and 10 tissue samples, respectively. However, double K-means is not of any help to understand which of the gene clusters discriminate the obtained partition of 21-10 tissue samples. In fact, although the two blocks formed by 707 up-regulated genes and partitioned into two clusters of 21 and 10 tissue samples are not biologically significant, they have not very close mean values. This consideration entitles us to believe that some of the 707 up-regulated genes

[1] In this way, the distance between two tissues is proportional to one minus the Pearson correlation coefficient between those tissues.

K	Q	BIC	AIC
2	2	375090	307532
3	2	373331	296104
4	2	372023	294985
4	4	374015	297061

Table 7.1: Double K-means results for Bittner et al. (2000) data

could be significant in explaining the tissue partition.

Figure 7.6: Best double K-means solution: K=4, Q=2

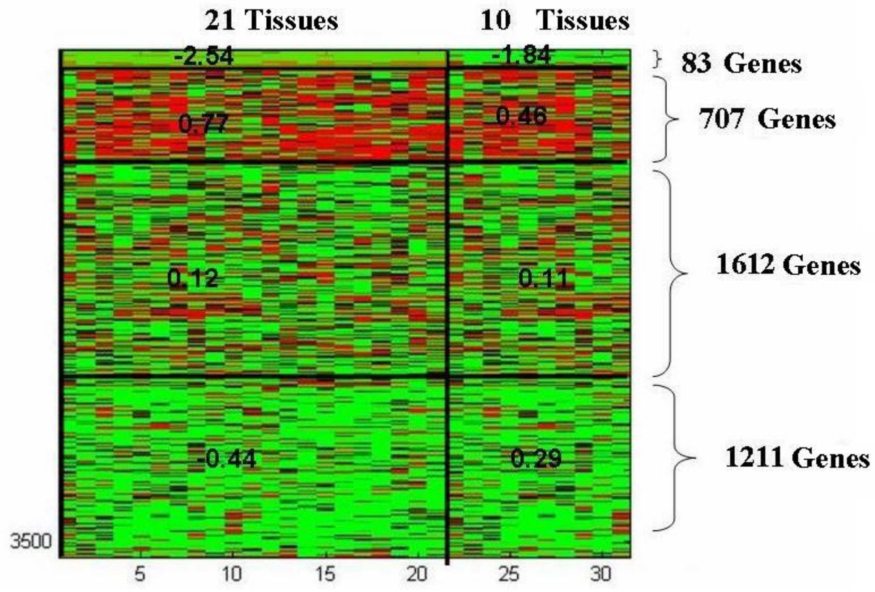
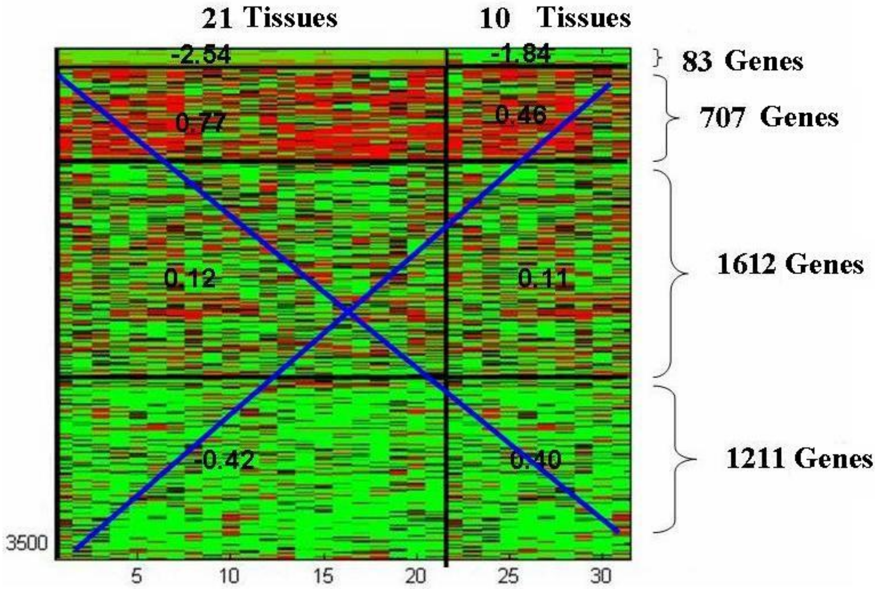


Figure 7.7: The significant blocks of best double K-means solution: K=4, Q=2



7.3.2 Double hierarchical mixture model

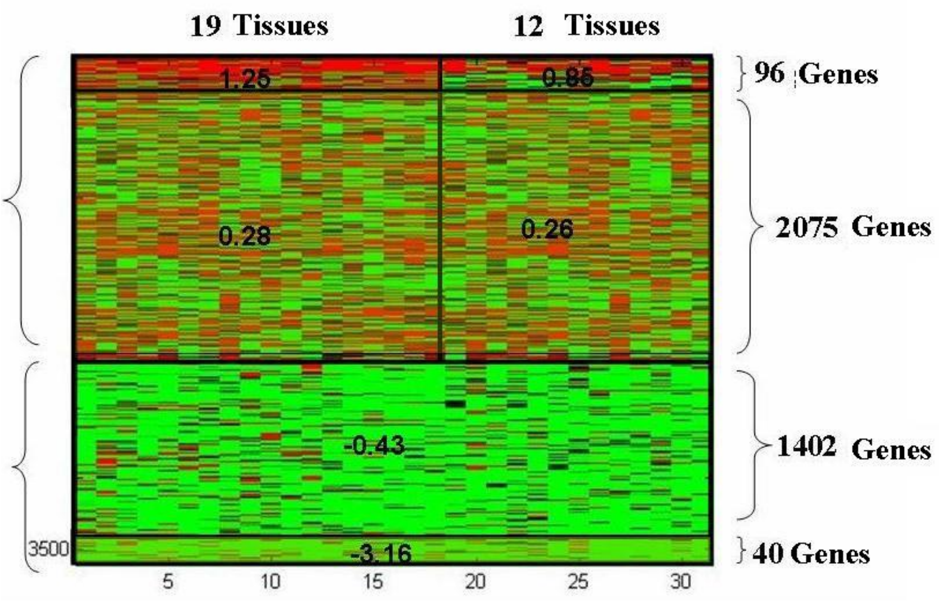
Also in this case, we estimated model parameters corresponding to different numbers of clusters for individuals and variables. For each combination, we run the upward-downward algorithm several times to avoid local minima, and recorded the solution giving the lowest value for BIC and AIC criteria. The results are summarized in Table 7.2.

The best solution is based on two 2nd level clusters, with two sub-clusters

K	T_k	Q_k	BIC	AIC
2	$T_1 = 3\ T_2 = 3$	$Q_1 = 2\ Q_2 = 1$	269780	264397
2	$T_1 = 2\ T_2 = 2$	$Q_1 = 2\ Q_2 = 2$	262260	249154
2	$T_1 = 2\ T_2 = 2$	$Q_1 = 2\ Q_2 = 1$	261720	248660
2	$T_1 = 2\ T_2 = 2$	-	261810	248730

Table 7.2: Double H-M model results for Bittner et al. (2000) data

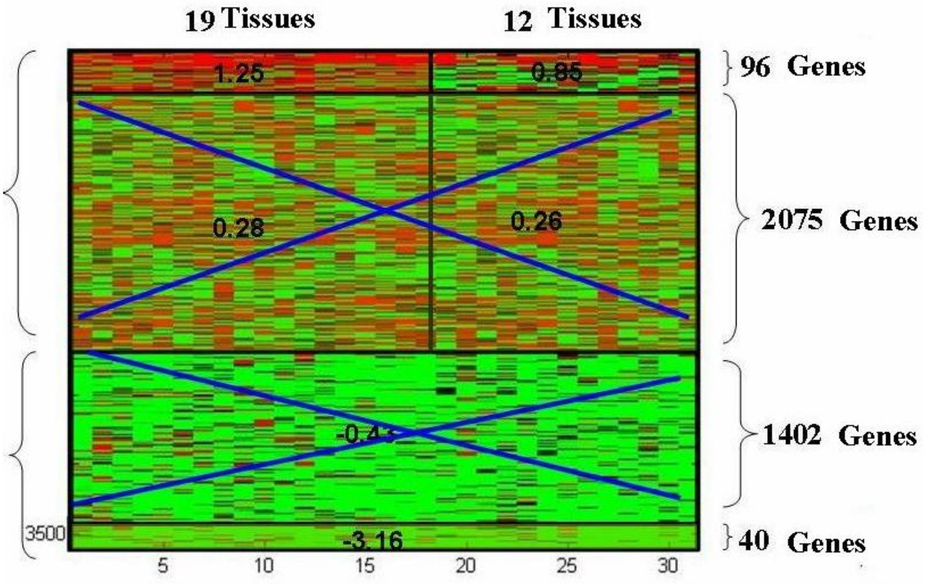
Figure 7.8: Best double H-M model solution: $K = 2$, $T_1 = T_2 = 2$, $Q_1 = 2$, $Q_2 = 1$



each. The column (tissue sample) cluster is based on a partition into two clusters for the former 2nd level cluster and on any partition for the latter 2nd level cluster (see line 3 in Table 7.1). In other words, the best solution corresponds to 6 different blocks, which are displayed in Figure 7.8. In particular, only 3 among those blocks have significant features (see Figure 7.9). In fact, the corresponding block means are far from zero. We obtained 96 up-regulated genes which contribute to determine a partition of tissue samples into 2 clusters of 12 and 19 samples each and 40 down-regulated genes with constant value over the analyzed tissue samples. We can claim that the 96 up-regulated genes are able to discriminate between the 19-12 tissue samples since the remaining 2075 genes, in the same 2nd level cluster, have very close mean values.

For each partition of tissue samples, we have a hierarchical partition of genes that could help highlight possible links between clusters of genes.

Figure 7.9: The significant blocks of best double H-M model solution: $K = 2$, $T_1 = T_2 = 2$, $Q_1 = 2$, $Q_2 = 1$



Thanks to the insertion of an extra-level, we could single out those genes clusters showing different functions or involved in different cellular processes.

Figures 7.10 displays the partition of tissue samples by using only the 96 up-regulated genes while Figure 7.11 displays the tissue samples block obtained by using only the 40 down-regulated genes. As it can be seen, any partition of tissue samples can be detected considering the down-regulated genes only, while a clear and distinct partition is obtained if we consider only the 96 up-regulated genes.

Figure 7.10: Partition of 12 and 19 tissue samples by using the 96 up-regulated genes (sample×average expression level)

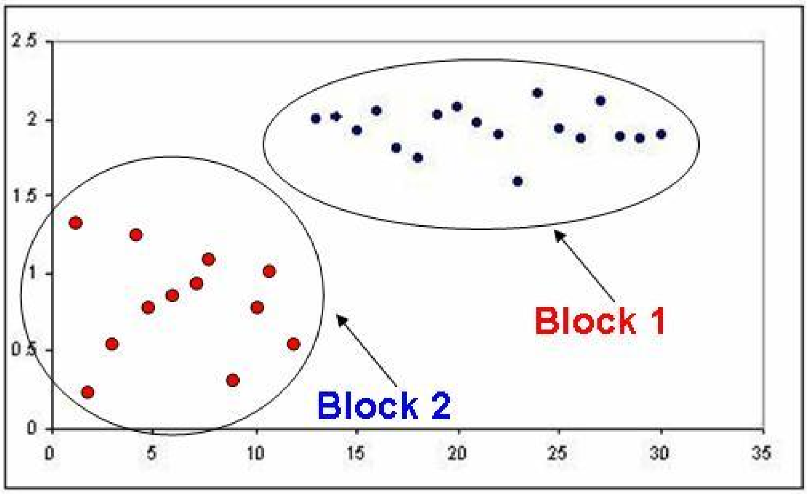
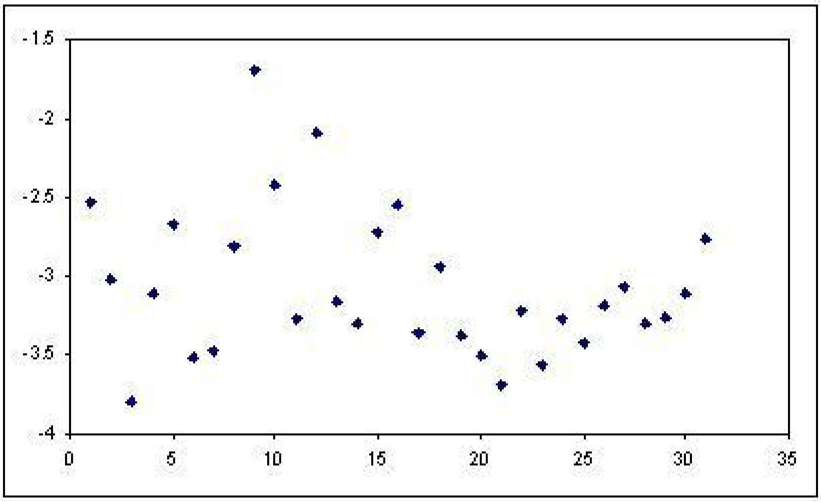


Figure 7.11: Block of tissue samples by using the 40 down-regulated genes (sample×average expression level)



7.3.3 Conclusions

We have tested the performance of proposed double clustering models on a benchmark data set from the study on cutaneous melanomas described in

Bittner et al. (2000). We applied the same pre-processing steps to allow for direct comparability with previously published analyses. Goldestein et al. (2002) applied several hierarchical clustering methods; Rocci and Vichi (2004) estimated the double K-means parameters using a least-squares approach. Both obtained a partition of tissue samples formed by 21 and 10 samples which agrees with that obtained by estimating the double K-means parameters through the maximum likelihood approach detailed in Section 5. Moreover, as far as gene clusters are concerned, Rocci and Vichi (2004) obtained 4 gene clusters with only one containing differentially expressed genes. Bittner et al. (2000), instead, obtained the same partition of 12-19 tissues samples we have obtained through the double H-M model, confirming that the two tissue sample clusters present different metastatic properties. By using the double H-M model we do not only obtain a partition of tissue samples in 2 groups with different metastatic properties, but also obtain a cluster composed by the up-regulated genes which seem highly significant in explaining the tissue partition.

7.4 Description of the Golub et al. data set (1999)

The data set described in Golub et al. (1999) comes from a study of gene expression in two types of acute leukemias: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Gene expression levels were measured using Affymetrix high-density oligonucleotide arrays (HU6800chip). Intensity values have been re-scaled such that the overall intensities for each array are equivalent (see http://www.broad.mit.edu/mpr/ publications/projects/ Leukemia/protocol.html).

The data set contains 7129 human genes from 72 tissue samples: 47 samples of acute lymphoblastic leukemia, ALL (38 hit in intermediate precursors of "lymphocytes" B, "B-cell", and 9 hit in intermediate precursors of "lymphocytes" T , "T-cell") and 25 samples of acute myeloid leukemia, AML. The 72 tissue samples have been divided in two sets: a training set containing 38 tissue samples, 27 ALL (8 T-cell and 19 B-cell) and 11 AML; a test set of 34 tissue samples, 20 ALL (1 T-cell and 19 B-cell) and 14 AML. In the training set only bone marrow samples are present, while the test set contains also peripheral blood samples.

Each sample has an associated gene expression level value and a corresponding "absolute call" (Present [P], Absent [A], Marginal [M]). The absolute calls are generated by the scanning software and give a categorical measure of the quality of expression values. In other words, the retro-transcribed gene can be actually Present/ Absent/ Marginal. However, it has been noted that Affymetrix software may not be sensitive to low gene expression values, and it may assess an "Absent call" even if the gene transcription has correctly occurred. In analyses of this data set, Golub et al. (1999) and McLachlan et al. (2002) ignored the absolute call information, and based their analysis only on gene expression levels. Aris and Recce (2002) have focused on genes that are selectively expressed rather than on differentially expressed genes. As a preprocessing step, it is necessary to adopt some transformation in order to reduce chip effects, background intensity, variations from RNA extraction, labelling, dye efficiency and other variability sources related to experiment. Several normalization methods have been proposed (see e.g. Alizadeh et al., 2000; Dopazo et al., 2001; Yeung et al., 2001a; Yeung et al., 2001b; Bolstard et al., 2003), but no "standard" method exists. We have applied the pre-processing scheme followed by McLachlan et al. (2002), Dudoit et al. (2002) and recommended by Pablo Tamayo, one of the authors of Golub et al. (1999). We used this scheme to allow for direct comparability with previous analyses on the same data set. The steps were:

1. Restrict gene expression values to the range (100, 16000). That is, gene expression levels above 16000 will be cut to 16000, since fluorescence saturation is present and corresponding values above this level cannot be reliably measured. Furthermore, gene expression levels below 100 will be set to 100.

2. Exclusion of genes with $(max/min) \leq 5$ or $(max - min) \leq 500$, where *max* and *min* refer to the maximum and minimum intensity for a particular gene across mRNA samples, respectively; this exclusion much of genes with reduced variability across samples.

3. Use base 10 logarithmic transformation.

4. Standardization of each column to have zero mean and unit variance followed by standardization of each row to have zero mean and unit variance. This was done to remove systematic sources of variation, as discussed in Dudoit and Fridlyand (2003).

7.5 Results

In the analysis of the Golub et al. (1999) data set, we do not consider the information provided by the absolute call and only analyzed the training set formed by 38 tissue samples and 3051 genes which have been selected after the pre-processing step previously described. We denote the gene expression levels by $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n$, where $\mathbf{y}_i$ is a 38-dimensional vector representing the expression level of the i-th gene on the 38 samples. Thus, we have $J = 38$ and $n = 3051$.

The goal associated is to classify malignancies into known classes (discriminant analysis) identifying homogeneously (over samples) expressed gene clusters.

7.5.1 Double K-means

We have fitted the double K-means model for different combinations number of clusters number for units (genes) and variables (tissue samples). For each combination, we run the algorithm2rid several times and choose the best solution by using the BIC and AIC criteria. Table 7.3 shows the obtained results. As it can be noticed, the best solution corresponds to 6 blocks defined by 3 genes clusters (with cardinality 946-1265-1254) and 2 tissue samples clusters (with cardinality 28-10) (see Table 7.3). The blocks are displayed in Figure 7.12. The block means close to zero could still have a significant biological meaning since the gene expression levels are absolute and not relative to some reference slide. As it can be seen from Figure 7.12, gene clusters are more clearly than those obtained in the previous data set; this can be probably due to the different technique adopted in the data acquisition process, where error sources causing extra design variation are well accounted for. As underlined before we are not able to establish which of the gene clusters is more informative with respect to tissue partition; however, the gene cluster formed by the 496 up-regulated genes seems to be the best candidate for 2 reasons:

1) usually the number of differentially expressed genes is small;

2) the corresponding block means formed are quite far away.

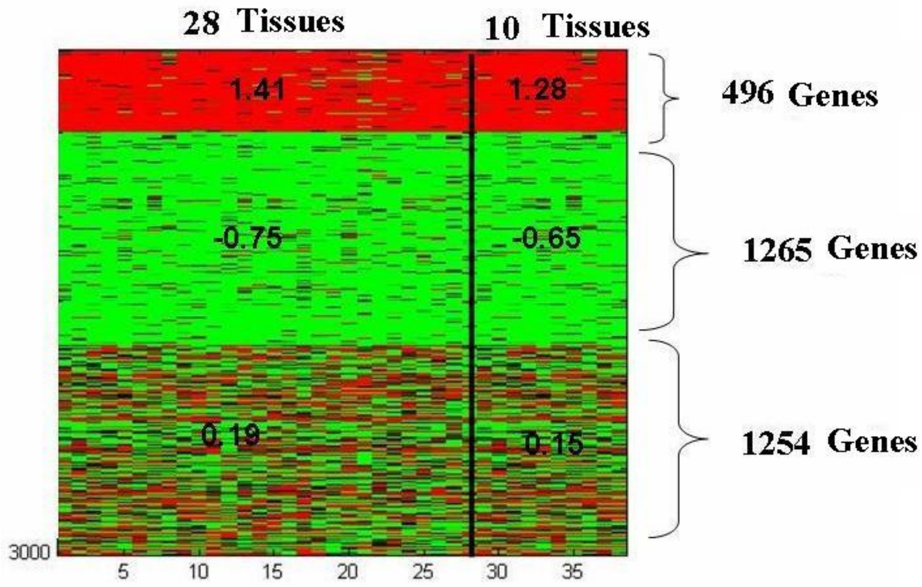
The partition of 28-10 tissue samples corresponds to the known classifi-

K	Q	BIC	AIC
2	2	270184	256978
3	2	268538	232104
4	2	269153	240587
2	3	290432	260391

Table 7.3: Double K-means results for Golub et al. (1999) data

cation in 2 group of ALL and AML tissue samples with a misclassification: AML sample n.66. However, this AML sample has been incorrectly classified also into other analyses (see e.g. Golub et al., 1999; Dudoit et al., 2002; Martella, 2006), so we can conclude that it is an outlying sample.

Figure 7.12: Best double K-means solution: K=3, Q=2



7.5.2 Double hierarchical mixture model

We have fitted the double H-M model for different combinations number of clusters for genes and tissues. We followed the same procedure detailed before using BIC retained the best solution (see Table 7.4) and obtained that displayed in Figure 7.13. Two 2nd level clusters with two subclusters each. The columns (tissue samples) are not partitioned in the former 2nd level cluster while a partition in three clusters for the latter 2nd level cluster is determined.

K	T_k	Q_k	BIC	AIC
2	$T_1 = 2$ $T_2 = 2$	$Q_1 = 2$ $Q_2 = 1$	199820	180538
2	$T_1 = 2$ $T_2 = 2$	$Q_1 = 3$ $Q_2 = 1$	199812	180529
2	$T_1 = 3$ $T_2 = 3$	$Q_1 = 2$ $Q_2 = 1$	221634	201432

Table 7.4: Double H-M model results for Golub et al. (1999) data

Through this model, we obtain a partition in 11-17-10 tissue samples corresponding to the right one. In particular, thanks to the hierarchical structure, we are also able to discriminate between B-cell and T-cell tissue samples (11-17), a hard task to deal with in one level double clustering models.

Figure 7.14 displays the obtained gene clusters. A related biological question would be to assess which of the 2nd level genes determine the partition: probably the 1103 genes corresponding to the 2nd level cluster are not all differentially expressed over tissue samples.

Figure 7.13: Best double H-M model solution: $K = 2$, $T_1 = T_2 = 2$, $Q_1 = 3$, $Q_2 = 1$

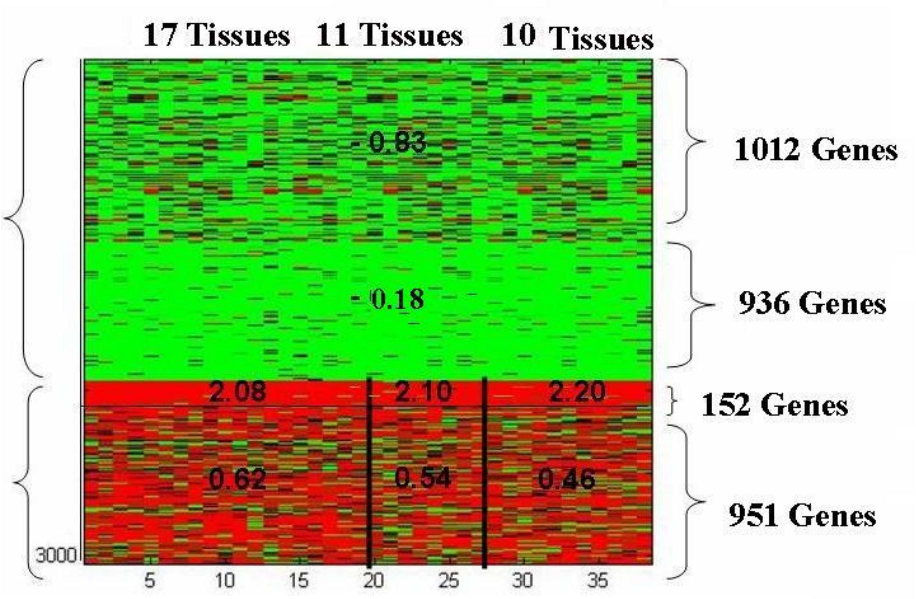
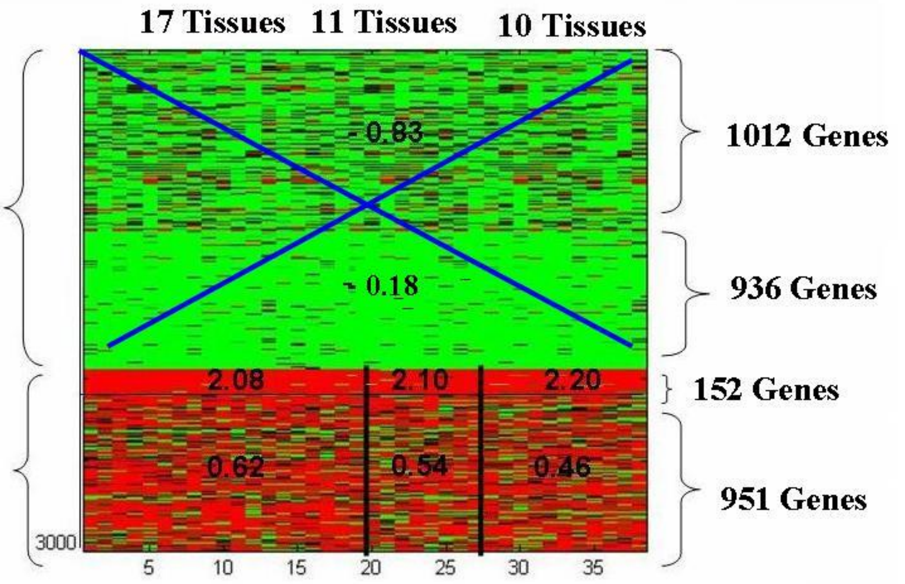


Figure 7.14: The significant blocks of best double H-M model solution: $K = 2$, $T_1 = T_2 = 2$, $Q_1 = 3$, $Q_2 = 1$



7.5.3 Conclusions

We discussed proposed methods in a context of "semi"-discriminant analysis; in fact, we have information about the tissue sample memberships and no information about gene clusters. We showed that our proposals have improved the standard results in term of classification of the 38 tissue samples (see i.e. Golub et al. 1999; Dudoit et al., 2002; McLachlan et al. 2002). In fact, we do not only obtain a partition of tissue samples in 2 known groups (ALL and AML) but, also by using the double H-M model, we recover the further partition of ALL tissue samples into T-cell and B-cell. However, which of the genes, among the 1103 selected by the latter model, contribute to tissue partition is still under study.

Chapter 8

Discussion and future work

The great amount of data, nowadays available, produces very often relevant problems for their analysis and interpretation. The traditional clustering techniques (non-supervised classification), that are often relevant in the data mining process, are not frequently suitable to the new application contexts, as for instance, text web mining, microarray and/or customer satisfaction. The dissertation focuses on the methodological development of data analysis techniques coming from these new application contexts, that are denoted as “high-multidimensional data”. Two main approaches to synthesize high-dimensional data have been discussed. Asymmetric (factorial reduction and clustering) and Symmetric (simultaneous clustering) approaches. Their use depends on the nature of analyzed data and the researcher’s task. Both approaches can be viewed under a not probabilistic or probabilistic framework. The latter has been developed to try solving important practical questions that arise in conventional clustering methods, such as the choice of the number of clusters or inference model parameters.

A critical review of standard K-means and model-based clustering techniques is provided. Then, we have discussed and compared models allowing for factorial reduction within a standard finite mixture model (see Ghahramani and Hinton, 1996; Rocci and Vichi, 2002).

Wide space has been devoted to describe double clustering methods; that is, approaches to clustering both units and variables. Many real fields could be mentioned where double clustering is meaningful and informative. For example, text/web mining, microarray data analysis, marketing and preference data analyses. We have focused on microarray analysis, where a major problem consists in clustering patients and tissues (in general, experimental conditions) which show similar behaviour with respect to genes expressions. In fact, many activation patterns are common to groups of genes only under specific experimental conditions. Therefore, the double clustering of rows and columns allows to achieve the further goal of detecting groups of genes with equivalent functions characterizing a specific subset of experimental conditions.

A schematic explanation of key principles underlying double clustering methods previously introduced in the literature has been given.

The rest of the dissertation deals with our proposal of model-based double clustering methods, whose effectiveness is highlighted by experimental comparisons on both simulated and gene expression data sets (discussed in Part II).

We extended the double K-means, introduced by Vichi (2000), to the probabilistic framework to reach less arbitrary criterion for selecting the number of clusters. An related issue is to extend this model to allow for different variables partitions in each unit cluster. Rocci and Vichi (2004) have generalized the double K-means to this purpose using a least squares approach.

Moreover, we proposed to adapt the multilevel latent class model of Vermunt (2003) to two-way continuous data. Observations are clustered into a particular (1st level) latent component within a certain (2nd level) cluster. In order to cluster variables we introduce a binary and row stochastic matrix of variable cluster membership (as in double K-means; Vichi, 2000). We have discussed a potential reparameterization of the (1st level) component-specific mean vector extending the work of Rocci and Vichi (2002). Thanks to the hierarchical structure, we learn the ground-truth data clusters by distinguishing the number of components (1st level components) from the number of clusters (2nd level components), which is often a problem in model-based clustering. We have discussed the identifiability for this model under specific constraints; the presented approach could be extended to more levels.

An interesting future research is in extending the work of Ghahramani and Hinton (1996) to allow for simultaneous clustering of units and variables; this can be done by using K factor models with mixtures of t distributions trying to make the model less sensitive to outliers (McLachlan et al., 2006).

Another encouraging research direction is on investigating how both models could be extended to categorical data and to define specific criteria to evaluate the discriminant power of unit clusters with respect to variable clusters.

Bibliography

[1] Akaike H. (1973). Information theory as an extension of the maximum likelihood principle. Pp. 267-281 in B. N. Petrov and F. Csaksi, editors. 2nd International Symposium on Information Theory. Akademiai Kiado, Budapest, Hungary.

[2] Alfó M., Martella F. and Vichi M. (2006). Double Hierarchical Mixture model for microarray data". IFCS (International Federation of Classification Societies), Invited Session on "Clustering and Classification of Microarray Gene Expression Data", Ljubljana, Slovenia (July, 2006).

[3] Alizadeh et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769): 503-511.

[4] Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D. and Levine A.J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array. *Proc. Natl. Acad. Sci.* USA, 96(12), 6745-6750.

[5] Amaratunga D. and Cabrera J. (2004). Exploration and Analysis of DNA Microarray and Protein Array Data. Wiley Series in Probability and Statistics.

[6] Ambler G. (2003). Bayesian two-way clustering for gene expression data, $http : //www.bgx.org.uk/presentations.html$.

[7] Aitkin M., Anderson D. and Hinde J. (1981). Statistical modeling of data on teaching styles (with discussion). *J. R. Statist. Soc.* A, 144, 419-461.

[8] Anderberg M.R. (1973). Cluster Analysis for Applications, New York: Academic Press, Inc.

[9] Anderson T.W. and Rubin H. (1956). Statistical inference in factor analysis. In: Proceedings of the third Berkeley symposium on mathematical statistics and probability (Vol. 5) (111-150). Berkeley.

[10] Arabie Ph. and Hubert L. (1994). Iterative Projection Strategies for the Least-squares Fitting of Graph Theoretic Structures to Proximity Data, *Research Report RR-94-02*, Department of Data Theory, University of Leiden; 62.

[11] Arabie P., Schleutermann S., Daws J. and Hubert L. (1988). Marketing applications of sequencing and partitioning of nonsymmetric and/or two-mode matrices. In W. Gaul and M. Schader, editors, Data Analysis, Decision Support, and Expert Knowledge Representation in Marketing, 215-24. Springer Verlag.

[12] Aris V. and Recce M. (2002). A method to improve detection of disease using selectively expressed genes in microarray data, *Methods of Microarray Data Analysis*, eds. SM Lin, KF Johnoson (Kluwer Academic), 69-80.

[13] Banfield J.D. and Raftery A.E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, 49, 803-821.

[14] Bartlett M.S. (1937). The statistical conception of mental factor, *British Journal of Psychology*, 28, 97-104.

[15] Bartlett M.S. (1938). Further aspects of the theory of multiple regression, *Proceedings of the Cambridge Philosophical society*, 34, 33-40.

[16] Ben-Dor A., Shamir R. and Yakhini Z. (1999). Clustering gene expression patterns. *Journal Comput Biol*, 6(3-4):281-97.

[17] Ben-Dor A., Chor B., Karp R., Yakhini Z. (2002). Discovering local structure in gene expression data: the order-preserving submatrix problem. In: Myers G., Hannenhalli S., Sankoff D., Istrail S., Pevzner P., Waterma, M. (Eds.), Proceedings of the Sixth Annual International Conference on Computational Biology (RECOMB-02). ACM Press, New York, NY, 49-57.

[18] Bittner et al. (2000). Molecular classification of cutaneous malignant by gene expression profiling, *Nature*, 406 (6795), 536-540.

[19] Bock H.H. (1974). Automatische Klassifikation, Vandenhoeck and Ruprecht, Gottingen.

[20] Bock H.H. (1996). Probability models and hypotheses testing in partitioning cluster analysis. In: Arabie P., Hubert L.J., De Soete G. eds. Clustering and classification. River Edge, Nj: World Scientific Publ..

[21] Bock H.H. (1998). Probabilistic aspects in classification. In: Ch. Hayashi et al. (ed.): Data science, classification and related methods. Springer-Verlag, Heidelberg, 3-21.

[22] Bolstad B.M., Irizarry R.A., Astrand M. and Speed T.P. (2003). A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance, *Bioinformatics*, 19 (2), 185-193.

[23] Boyles R. (1983). On the convergence of the EM algorithm. *JRSS B*, 47-50.

[24] Bozdogan H. and Sclove S.L. (1984). Multi-Sample Cluster Analysis using Akaike 's Information Criterion. *Annals of Institute of Statistical Mathematics*, 36, 163-180.

[25] Breiman L., Freidman J.H., Olshen R.A., Stone C.J. (1984). Classification and Regression Trees. Wadsworth.

[26] Brusco M.J. and Cradit J.D. (2001). A variable selection heuristic for k-means clustering. *Psychometrika* 66, 249-270.

[27] Busygin S., Jacobsen G. and Kramer E. (2002). Double conjugated clustering applied o leukemia microarray data. In *Proceedings of the 2nd SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data.*

[28] Califano A., Stolovitzky G. and Tu Y. (2000). Analysis of gene expression microarrays for phenotype classification. *Proc Int Conf Intell Syst Mol Biol*, 8:75-85.

[29] Celeux G., Chauveau D. and Diebolt J. (1996). Stochastic versions of the EM algorithm: an experimental study in the mixture case. *J. of Statist. Comput. Simul.*, 55, 287-314.

[30] Celeux G. and Diebolt D. (1985). The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational Statistics Quarterly*, 2(1): 73-82.

[31] Celeux G. and Diebolt J. (1992). A stochastic approximation type EM algorithm for the mixture problem, *Stoch. and Stoch. Reports*, 41, 119-134.

[32] Celeux G. and Govaert G. (1992). A Classification EM Algorithm for Clustering and Two Stochastic versions. *Computational Statistics and Data Analysis*, 14, 315-332.

[33] Celeux G. and Govaert G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781-93.

[34] Celeux G. and Soromenho G. (1996). An Entropy Criterion for Assessing the Number of Clusters in a Mixture Model. *Classification Journal*, 13, 195-212.

[35] Chang W. (1983). On using principal components before separating a mixture of two multivariate normal distribution. *Applied Statistics*, 32, 267-275.

[36] Chauveau D. (1995). A stochastic EM algorithm for mixtures with censored data. *J. Statist. Plann. Inference*, 46 (1), 1-25.

[37] Cheng Y. and Church G.M. (2000). Biclustering of expression data. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 8:93 103.

[38] Chickering D. Maxwell and Heckeman D. (1997). Efficient Approximations for the Marginal Likelihood of Bayesian Networks with Hidden Variables. *Machine Learning*, 29, 181-212.

[39] Cheeseman P. and Stutz J. (1995). Bayesian classication (AutoClass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*, 153-180.

[40] Cho H., Dhillon I.S., Guan Y. and Sra S. (2004). Minimum Sum-Squared Residue Co-clustering of Gene Expression Data. *Proceedings of the Fourth SIAM International Conference on Data Mining*, 114-125.

[41] Dasgupta A. and Raftery A.E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93, 294-302.

[42] Dempster A.P., Laird N.M., Rubin D.A. (1977). Maximum likelihood from incomplete datavia the EM Algorithm. *Journal of the Royal Statistical Society.* B, 39, 1-38.

[43] DeSarbo W.S. (1982). GENNCLUS: new models for general nonhierarchical clustering analysis. *Psychometrika*, 47: 449-475.

[44] Desarbo W.S., Carroll J.D. , Clarck L.A. and Green P.E. (1984). Synthesized clustering: A method for amalgamating clustering bases with differential weighting of variables. *Psychometrika* 49, 57-78.

[45] DeSarbo W.S., Kamel Jedidi, Karen Cool and Dan Schendel (1990). Simultaneous multidimensional unfolding and cluster analysis: An investigation of strategic groups. *Marketing Letters*, 3, 129-146.

[46] De Soete G. and Carroll J.D. (1994). k-means clustering in a low-dimensional Euclidean space. In E. Diday, Y. Lechevallier, M. Schader,P. Bertrand and B. Burtschy (Eds.), *New Approaches in Classification and Data Analysis*, 212-219. Berlin: Springer-Verlag.

[47] De Soete G. and Carroll J.D. (1996). TreeCand other network models for representing proximity data. In P. Arabie, L. J. Hubert, and G. De Soete (Eds.), *Clustering and Classification*, 157-197. River Edge, NJ: World Scientific Publishing.

[48] Devaney M. and Ram a. (1997). Efficient feature selection in conceptual clustering. In Machine Learning: *Proceedings of the Fourteenth International Conference*, Nashville, TN, 92-97.

[49] Dhillon I.S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In: Provost, F., Srikant, R. (Eds.), Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining. ACM Press, New York, NY, 269-274.

[50] Dias R. and Gamerman D. (2002). A Bayesian approach to hybrid splines non-parametric regression, *Journal of Statistical Computation and Simulation*, 72(4), 285-297.

[51] Diebolt J. and Celeux G. (1993). Asymptotic Properties of a Stochastic EM Algorithm for Estimating Mixing Proportions. *Stochastics Models*, 9, 599-613.

[52] Diebolt J. and Robert C.P. (1994). Estimation of finite mixture distributions through Bayesian sampling. *J. Roy. Statist. Soc.* B, 56, 363-375.

[53] Dillon W.R. and Mulani N. (1989). LADI: A latent discriminant model for analyzing marketing research data. *Journal of Marketing Research*, 26: 15-29.

[54] Di Zio M., Guarnera U., Luzi O. (2005). Editing systematic unity measure errors through mixture modelling. *Surv. Methodol.* 31, 53-63.

[55] Dopazo J., Zanders E., Dragoni I., Amphlett G. and Falciani F. (2001). Methods and approaches in the analysis of gene expression data, *J. Immunol. Meth.*, 250, 93-112.

[56] Dudoit S. and Fridlyand J. (2003). Classification in microarray experiments, *Analysis of microarray experiments*, Chapman and Hall-CRC, edited by T. P. Speed.

[57] Dudoit S., Fridlyand J. and Speed T.P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data, *Journal of the American Statistical Association*, 97, 457, 77-87.

[58] Eckes T. and Orlik P. (1993). An error variance approach to two-mode hierarchical clustering. *Journal of Classification*, 10: 51-74.

[59] Eisen et al. (1998). Clustering analysis and display of genom-wide expression patterns.*Proc. of Nat. Acad. of Sci.*, USA, 95, 14863-14868.

[60] Feng Z.D. and McCulloch C.E. (1996). Using bootstrap likelihood ratios in finite mixture models. *J. R. Statist. Soc.* B, 58(3), 609-617.

[61] Fessler J.A. and Hero A.O. (1994). Space-alternating generalized EM algorithm.*IEEE Trans. on Signal Proc*, 42, 2664-2677.

[62] Fisher W. (1969). Clustering and aggregation in economics. *Baltimore: Johns Hopkins.*

[63] Fowlkes E.B. (1979). Some Methods for Studying the Mixture of Two Normal (Lognormal) Distributions. *Journal of the American Statistical Association*, 74, 367, 561-575.

[64] Fraley C. and Raftery A.E. (1998). How Many Clusters? Which Clustering Method? Answers Via Model Based Cluster Analysis. *Computer Journal*, 41, 578-588.

[65] Friedman J.H., Meulman J.J. (2004). Clustering objects on subsets of attributes. *J. R. Statist. Soc.* B 66 (4), 815849.

[66] Getz G., Levine E. and Domany E. (2000). Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci.* USA, 97(22), 12079-12084.

[67] Getz G., Gal H., Kela I., Notterman D. A. Domany, E. (2003). Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data. *Bioinformatics* 19 (9), 1079-1089.

[68] Ghahramani Z. and Hinton G.E. (1996). The EM algorithm for mixture of factor analyzers. *Technical Report*, CRG-TR-96-1, 8, University of Toronto.

[69] Goldstein D., Ghosh D. and Conlon E. (2002). Statistical issues in the clustering of gene expression data. *Stat Sinica*, 12:219241.

[70] Golub T.R. et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531-537.

[71] Good E.E. (1952). The life history of the American Crow-Corvus bruchyrhynchos Brehm. Ph.D. diss., Ohio State Univ., Columbus, OH.

[72] Gordon A.D. (1999). Classification (2nd edition). *Chapman and Hall/CRC*, Boca Raton.

[73] Govaert G. (1980). Classification croise de tableaux de contingence. In: Premires journs internationales analyse des donnees et informatique (Versailles 1977). Paris: CNRS.

[74] Govaert G., Nadif M. (2003). Clustering with block mixture models. *Pattern Recognition* 36(2): 463-473.

[75] Hartigan J.A. (1972). Direct clustering of a data matrix. *J. Am. Stat. Assoc.*, 67, 123-129.

[76] Hartigan J.A. (1975). Clustering algorithms. New York: York Wiley.

[77] Hastie T. and Tibshirani R. (1996), Discriminant analysis by Gaussian mixtures. *J. Roy. Statist. Soc. Ser. B* 58, 155-176.

[78] Hathaway R. (1986). Another interpretation of the EM algorithm for mixture distributions, *Statistics and Probability Letters*, 4, 53-56.

[79] Hope A.C.A. (1968). A simplified Monte Carlo significance test procedure. *Journal of the Royal Statistical Society*, series B, 30:582-598.

[80] Hubert L. and Arabie P. (1985). Comparing partitions. *Journal of Classification*, vol. 2, 193-218.

[81] Ichihashai H., Honda K., Tani N. (2000). Gaussian mixture pdf approximation and fuzzy c-means clustering with entropy regularization. *Proc. of the 4th Asian Fuzzy System Symp.*, Tsukuba, Japan, 31 May-3 June, 217-221.

[82] Initial sequencing and analysis of the human genome (2001). International Human Genome Sequencing Consortium. *Nature 409*, 942-943.

[83] Johnson S.C. (1967). Hierarchical Clustering Schemes. *Psychometrika*, 2:241-254.

[84] Jolliffe I.T., Jones B. and Morgan B.J. (1980). cluster analysis of the elderly at home: a case study. *Data analysis and Informatics*, 745-757.

[85] Kass R.E. and Raftery A.E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90, 773-795.

[86] Kluger Y., Basri R., Chang J.T., Gerstein M. (2003). Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res.* 13 (4), 703- 716.

[87] Kullback S. and Leibler R.A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22:79-86.

[88] Lavine M. and West M. (1992). A Bayesian method for classification and discrimination. The Canadian Journal of Statistics 20, 451-461.

[89] Lazzeroni L., Owen A., (2002). Plaid models for gene expression data. *Statist. Sinica*, 12 (1), 61-86.

[90] Lehmann E.L. (1980). Efficient likelihood estimators. *The American Statistician*, 34(4):2 :233-235.

[91] Leroux B.G. (1992). Maximum-likelihood estimation for hidden Markov models. *Stochastic Processes, Applicat.*, 40, 127-143.

[92] Li J. (2005). Clustering based on a multi-layer mixture model, *Journal of Computational and Graphical Statistics*, 14(3), 547-568.

[93] Li Leping, Weinberg Clarice R., Darden Thomas A. and Pedersen Lee G. (2001). Gene selection for sample classification based on gene expression

data: study of sensitivity to choice of parameters of the ga/knn method. *Bioinformatics*, 17, 1131-1142.

[94] Liu C. and Rubin D.B. (1994). The ECME Algorithm: A Simple Extension of EM and ECM with Faster Monotone Convergence. *Biometrika* 81, 633-648.

[95] Liu J. and Wang W. (2003). Op-cluster: Clustering by tendency in high dimensional space. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, 187-194.

[96] MacKay D.J.C., Miskin J. (2001). Latent variable models for gene expression data. Tech. rep., Cavendish Lab., Cambridge Univ., UK, http://www.inference.phy. cam.ac.uk/mackay/abstracts/icagenes.html.

[97] Madeira Sara C. and Oliveira Arlindo L. (2004). Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1, 24-45.

[98] Martella F. (2006). Classification of microarray data with factor mixture models. *Bioinformatics* 22(2): 202-208.

[99] Martella F. and Vichi M. (2006). Model-based double clustering. Atti del convegno XX RIUNIONE SCIENTIFICA, Rome (19th-20th January) and submitted.

[100] McCallum A., Nigam K. and Ungar L. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 169-178.

[101] McLachlan G.J. (1987). On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture, *Appl. Statist.*, 36, 318-324.

[102] McLachlan G.J. and Basford K.E. (1988). Mixture Models: Inference and Applications to Clustering. *Marcel Dekker*, New York.

[103] McLachlan G.J., Basford K., and Bean R. (2006). Issues of robustness and high dimensionality in cluster analysis. Atti del convegno di COMPSTAT2006, Universitá degli Studi di Roma La Sapienza, (28th August-1th September).

[104] McLachlan G.J., Bean R.W., and Ben-Tovim Jones L. (2006). Extension of the mixture of factor analyzers model to incorporate the multivariate t distribution. Computational Statistics and Data Analysis. To appear.

[105] McLachlan G.J., Bean R.W. and Peel D. (2002). A mixture model-based approach to the clustering of microarray expression data, *Bioinformatics*, 18 (3), 413-422.

[106] McLachlan G.J. and Krishnan T. (1997). The EM Algorithm and Extensions. New York: Wiley.

[107] McLachlan G.J. and Peel D. (1997). On a Resampling Approach to Choosing the Number of Components in Normal Mixture Models.*In Computing Science and Statistics* 28, L. Billard and N.I. Fisher (Eds.). Fairfax Station, Virginia: Interface Foundation of North America, 260-266.

[108] McLachlan G.J. and Peel D. (2000a). Finite Mixture Models. *Wiley*, New York.

McLachlan G.J. and Peel D.(2000b). Mixtures of factor analyzers. In: Langley, P. (ed) Proceedings of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco.

[109] McLachlan G.J., Peel D., Basford K.E. and Adams, P. (1999). The EMMIX software for the fitting of mixtures of normal and t-components. *Journal of Statistical Software* 4, No. 2.

[110] McQueen J B. (1967). Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297.

[111] Meng X.L. and Rubin D.B. (1991). Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm. *Journal of the American Statistical Association*, 86:899-909.

[112] Meng X.L. and Rubin D.B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* 80: 267-278.

[113] Meng X.L. and Van Dyk D.A. (1997). The EM algorithm -an old folk song sung to a fast new tune. Reading paper in *J. R. Statist. Soc.*, Ser. B, 59, 511-567.

[114] Mickey M.R. et al. (1983). Boolean factor analysis. In: Dixon W.J. ed. BMDP Statistical Software Manual. Berkeley, CA: University of California Press.

[115] Mirkin B. et al. (1995). Additive two-mode clustering: the error-variance approach revisited. *Journal of Classification*, 12: 243-263.

[116] Murali T.M. and Kasif S. (2003). Extracting conserved gene expression motifs from gene expression data. In *Proceedings of the Pacific Symposium on Biocomputing*, 8, 77-88.

[117] Murtagh F. and Raftery A.E. (1984). Fitting straight lines to point patterns. *Pattern Recognition*, 17, 479-483.

[118] Pearl J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA.

[119] Pollard K.S. and van der Laan M.J. (2002). Statistical Inference for Simultaneous Clustering of Gene Expression Data, revised for Mathematical Biosciences: 176(1), 99-121.

[120] Quackenbush J. (2001). Computation analysis of microarray data. *Nature Reviews Genetics*, 2, 418-427.

[121] Ramsay J.O. (2003). Matlab, rands-plus functions for Functional Data Analysis, $ftp://ego.psych.mcgill.ca/pub/ramsay/FDAfuns$.

[122] Ramsay J.O.and Li X. (1998). Curve registration, *J.R. Stat. Soc. Ser. B Stat. Methodol.*, 60 (2), 351-363.

[123] Ramsay J.O. and Silverman B.W. (2002). Applied functional data analysis, Springer Series in Statistics, Springer-Verlag, New York. Methods and case studies.

[124] Rice J.A. (2000). Personal communication.

[125] Rocci R. and Vichi M. (2002). A two-way model for simultaneous reduction and classification, Atti della XLI Riunione Scientifica, Università di Milano-Bicocca (5th-7th June).

[126] Rocci R., Vichi M. (2004). Multimode partitioning, submitted.

[127] Rubin D.B. and Thayer D.T. (1983). EM algorithms for ML factor analysis. *Psychometrika*, 47, 69-76.

[128] Schwarz G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics,* 6, 2, 461-464.

[129] Sclove S.L. (1987). Application of model-selection criteria to some problems in multivariate analysis. *Psychmetrika*, 52: 333-343.

[130] Segal E., Battle A., Koller D. (2003). Decomposing gene expression into cellular processes. In: Pacific Symposium on Biocomputing. Vol. 8. $http://helix-web.stanford.edu/psb03/$, 89-100.

[131] Segal E., Taskar B., Gasch A., Friedman N., Koller D. (2001). Rich probabilistic models for gene expression. *Bioinformatics*, 17 (Suppl. 1), S243-S252.

[132] Sheng Q., Moreau Y. and De Moor B. (2003). Biclustering micrarray data by gibbs sampling. *Bioinformatics*, 19 (Suppl. 2), 196-205.

[133] Shental N., Bar-Hillel A., Hertz T. and Weinshall D. (2003). Computing mixture models with EM using side information (Equivalence Constraints). In The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, Workshop in ICML2003.

[134] Smyth P. (2000). Model selection for probabilistic clustering using cross-validated likelihood *Statistics and Computing*, 9, 63-72.

[135] Soromenho G. (1993). Comparing approaches for testing the number of components in a finite mixture model. *Computational Statistics*, 9, 65-78.

[136] Tamayo P., Slonim D., Mesirov J., Zhu Q., Kitareewan S., Dmitrovsky E., Lander E. and Golub T. (1999). Interpreting patterns of gene expression with self-organizing maps. *PNAS*, 96:2907-2912.

[137] Tanay A., Sharan R., Shamir R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 (Suppl. 1), S136-S144.

[138] Tang Chun, Zhang Li, Zhang Idon and Ramanathan Murali (2001). Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering, 41-48.

[139] Thomson G.H. (1951). The factor analysis of human ability (5th ed.) Boston: Houghton Mifflin Company.

[140] Tipping M.E., Bishop C.M. (1997). Mixtures of probabilistic principal component analysers. Technical Report, Neural Computing Research Group, Aston University.

[141] Titterington D., Smith A., Makov U (1985). Statistical Analysis of Finite Mixture Distributions. Wiley.

[142] Tryon R.C. (1939). Cluster Analysis. *Edwards Brothers.*

[143] Van Dyk D.A., Meng X.L., Rubin D.B. (1995). Maximum likelihood estimation via the ECM algorithm: computing the asymptotic variance. *Statistica Sinica* 5, 55-75.

[144] Van Mechelen I. (2006). Biclustering Methods for Microarray Gene Expression Data: Towards a Unifying Taxonomy. Data Science and Clas-

sification 10th Jubilee Conference of the International Federation of Classification Societies Program and Abstracts July 25-29, Ljubljana, Slovenia.

[145] Van Mechelen I. and Schepers J. (2006). A unifying model for biclustering. Atti del convegno di COMPSTAT2006, Universitá degli Studi di Roma La Sapienza, (28 August-1 September).

[146] Van Mechelen I., Bock H.H., De Boeck P. (2004). Two-mode clustering methods: A structured overview. Statistical Methods in medical Research 13: 363-394.

[147] Venter J.C. et al. (2001). The sequence of the human genome. *Science 291*, 1304-1351.

[148] Vermunt J.K. (2003). Multilevel latent class models. *Sociological Methodology* 33, 213-239.

[149] Vichi M. (2000). Double k-means Clustering for simultaneous classification of Objects and Variables. In Borra et al. (eds): Advances in Classification and Data Analysis, 43-52, Springer.

[150] Vichi M. and Kiers H.A.L. (2001). Factorial k-means analysis for two-way data. *Computational Statistics and Data Analysis*, 37, 49-64.

[151] Wang H., Wang W., Yang J. and Yu Philip S. (2002). Clustering by pattern similarity in large data sets. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, 394-405.

[152] Wei G. and Tanner M. (1990). A monte-carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411): 699-704.

[153] Willse A. and Boik R.J. (1999). Identifiable finite mixtures of location models for clustering mixed-mode data. *Statist. Comput.* 9, 111-121.

[154] Wolfe J.H. (1963). Object cluster analysis of social areas. Master's thesis, University of California, Berkeley.

[155] Wu C.F.J. (1983). On the Convergence Properties of the EM Algorithm. *Annals of Statistics*, 11, 95-103.

[156] Yang J. et al. (2002). δ-Clusters: Capturing Subspace Correlation in a Large Data Set. *Proc. 18th IEEE Int'l Conf. Data Eng.*, 517-528.

[157] Yang J. et al. (2003). Enhanced Biclustering on Expression Data. Proc. Third IEEE Conf. Bioinformatics and Bioeng., 321-327.

[158] Yeung K.Y., Fraley C., Murua A., Raftery A.E. and Ruzzo W.L. (2001a). Model-based clustering and data transformations for gene expression data, *Bioinformatics*, 17, 977-987.

[159] Yeung K.Y., Fraley Chris, Muru Alejandro, Raftery Adrian E. and Ruzzo Walter L. (2001b). Model-based Clustering and Data Transformations for Gene Expression Data, *Supplementary Web Site to Bioinformatics*, 17, 977-987.

[160] Zhao Qi and Miller D.J. (2005). Mixture Modeling with Pairwise, Instance-Level Class Constraints. *Neural Comp.*, 17: 2482-2507.

Acknowledgments

Dedico questo scritto a due persone a me care, mia sorella Cecilia e mio zio Pino, con cui avrei avuto il piacere di condividere questo momento. Sono sicura che se sono arrivata ai ringraziamenti di questo lavoro è anche grazie a voi.

Ringrazio particolarmente i miei genitori per aver sopportato i momenti di crisi e per avermi sostenuto sempre in questi anni di studio, spero di avere dato anche a loro la soddisfazione che si meritano.

Un sentito ringraziamento è per il mio supervisore, Maurizio Vichi, per avermi suggerito l'argomento ed essermi stato vicino nella stesura della tesi, interessandosi professionalmente ed umanamente anche nei momenti di difficoltà.

Un grazie di cuore è per Marco Alfò per avermi trasmesso la passione per la ricerca, per i suoi utili suggerimenti, per essere stato un riferimento fondamentale nel corso di questi anni e aver contribuito significativamente alla rivisitazione di questo lavoro, ma soprattutto per aver mostrato fiducia in me spronandomi sempre ad andare avanti.

Ringrazio Luciano Nieddu e Donatella Vicari per la loro disponibilità, per i frequenti confronti e per i numerosi suggerimenti.

Desidero inoltre ringraziare il Prof. R. Coppi per l'ottima attività di coordinamento del ciclo di dottorato ed, in generale, tutte le persone all'interno del Dipartimento con cui ho avuto il piacere di interagire in questi anni.

Ringrazio tutti i colleghi di dottorato e gli amici di sempre...evito di fare una lista per paura di dimenticarne qualcuno...grazie per essermi stati vicini in questo cammino curando le ferite e brindando ai festeggiamenti!

Con un pò di malinconia mista a timore e profonda curiosità verso il domani...concludo questo lavoro con la frase che segue...

"Ogni processo di conoscenza è come un mosaico e ciascun gradino successivo lascia sempre dietro di sè qualcosa di irrisolto. Così è anche la vita. È infantile pretendere di attraversarla protetti in ogni momento da certezze immutabili."