Chapter 1

Cluster Analysis

1.1. Introduction

Cluster analysis or clustering, which is an important tool in a variety of scientific areas including pattern recognition, information retrieval, microarrays and data mining, is a family of exploratory data analysis methods that can be used to discover structures in data. These methods seek to obtain a reduced representation of the initial data and, along with principal component analysis, factor analysis and multidimensional scaling, are one form of data reduction. The aim of cluster analysis is the organization of the set into homogeneous classes or natural classes, in a way which ensures that objects within a class are similar to one another. For example, in statistics, cluster analysis can identify several populations within a heterogeneous initial population, thereby facilitating a subsequent statistical study; in natural science, the clustering of animal and plant species, first proposed by Linnaeus (an 18th-Century Swedish naturalist), is a famous example of cluster analysis; in the study of social networks, clustering may be used to recognize communities within large groups of people and, at a more
Co-Clustering

general level, simply naming of objects can be seen as a form of clustering.

The attempt to formally define clustering, as a basis for an automated process, raises a number of questions. How can we define the objects (elements, cases, individuals or observations) to be classified? What is a cluster? How are clusters structured? How can different partitions be compared? Most often, the first step consists of defining the notion of proximity, a measure of closeness that can be similarity, dissimilarity or distance, among the objects to be clustered: two objects are close when their dissimilarity or distance is small or their similarity is large. Sometimes these proximities are the form in which the data naturally occur. In most clustering problems, however, each of the objects under investigation will be described by a set of variables or attributes, and the first step, possibly the most important, in clustering is to define these proximities. Then, a numerical function, usually known as a criterion, measuring the homogeneity of the clusters must be defined.

A classical example of a criterion used when the objects \(x_1, \ldots, x_n\) are described by \(d\) continuous variables is the within-group sum of squares, also called within-group inertia. In this situation, each individual being characterized by a vector \(x_i = (x_{i1}, \ldots, x_{id})\), the data take the form of a matrix \(X\) of dimension \((n, d)\) defined by values \(x_{ij}\), where \(i\) belongs to a set \(I\) of \(n\) observations and \(j\) belongs to a set \(J\) of \(d\) continuous variables. The within-group sum of squares can therefore be written as

\[
I_W(z) = \frac{1}{n} \sum_{i,k} z_{ik} d^2(x_i, \overline{x}_k) = \frac{1}{n} \sum_{i,k} z_{ik} ||x_i - \overline{x}_k||^2, \tag{1.1}
\]

where \(\overline{x}_k\) is the mean vector of the \(k\)th cluster and \(d\) is the Euclidean distance. Using the within-group covariance matrix
\( S_W = \frac{1}{n} \sum_k z_k S_k \), where \( z_k \) is the size of the \( k \)th cluster and \( S_k \) is the covariance matrix of the \( k \)th cluster

\[
S_k = \frac{1}{z_k} \sum_i z_{ik} (x_i - \bar{x}_k)(x_i - \bar{x}_k)^t,
\]

this criterion can also be written as \( I_W(z) = \text{trace}(S_W) \). The closer the within-group sum-of-squares criterion is to 0, the more homogeneous the partition will be. In particular, this criterion will be equal to 0 for a partition where each object is a cluster.

The problem can then appear very simple: from the finite set of partitions, select the partition that optimizes the numerical criterion. Unfortunately, the number of partitions is too large for them to be enumerated in a realistic time frame, because of combinatorial complexity. Generally, heuristics are used that, rather than giving the best solution, give a “good” solution close to the optimal solution and lead to local optimization. For instance, the two most commonly used clustering algorithms, namely the \( k \)-means algorithm for obtaining partitions and Ward’s hierarchical clustering method for obtaining hierarchies, use the within-group sum of squares criterion \( \text{trace}(S_W) \) derived from the within-group covariance matrix \( S_W \), and used the Euclidean metric as a measure of proximity.

In recent years, what used to be an algorithmic, heuristic and geometric focus has tended to give way to a more statistical approach using probabilistic clustering models to formalize the intuitive notion of a natural class [BOC 89]. This approach allows precise analysis and can provide a statistical interpretation of certain metrical criteria whose different variants are not always clear (such as the within-group sum of squares criterion \( \text{trace}(S_W) \)), as well as yielding new variants corresponding to precise hypotheses. It also represents a formal framework for tackling difficult
problems such as determining the number of classes or validating the obtained clustering structure. We should bear in mind that in many cases the set to be segmented is merely a sample drawn from a much larger population, and that the conclusions drawn from clustering the sample are to be extrapolated to the entire population. Here, clustering becomes meaningless in the absence of a probabilistic model justifying this extrapolation. All probabilistic approaches to clustering first assume that the data represent a random sample $x_1, \ldots, x_n$ from among a population, and then use an analysis of the probability distribution of this population to define a clustering. A number of different probabilistic clustering methods have been proposed, but the most traditional approach is the use of mixture models, which forms the main subject of this chapter.

Section 1.2 presents a brief review of the main approaches to clustering. Sections 1.3 and 1.4 deal with, respectively, the probability mixture models and the EM algorithm, the standard tool for estimating the parameters of such models. Section 1.5 describes how clustering may be carried out using a mixture model. The four subsequent sections describe several classical situations including Gaussian mixture models for continuous variables, and the latent class model for binary variables, categorical variables and contingency table, and in section 1.10, we study the implementation of these different methods.

1.2. Miscellaneous clustering methods

1.2.1. Hierarchical approach

In this section, it will generally be assumed that all the relevant relationships within the set to be classified are summarized by a dissimilarity $d$. The aim of hierarchical methods is to construct a sequence of partitions of a set
varying from partitions of singletons to the whole set. There
are two principal approaches. The divisive approach starts
with just one cluster containing all the objects. In each
successive iteration, clusters are split into two or more
further clusters, usually until every object is alone within a
cluster. Note that other stop conditions can be used, and the
division into clusters is governed by whether or not a
particular property is satisfied. For example, in taxonomy,
animals may be separated into vertebrates and invertebrates.
The agglomerative approach, in contrast to the divisive
approach, starts out from a set of \( n \) clusters, with each object
forming a singleton cluster. Then, in each successive
iteration, the closest clusters are merged until just one
cluster remains. Using a dissimilarity \( D \) among groups, the
closest clusters are the two clusters that are the closest with
respect to \( D \). According to the definition of \( D \), several
agglomerative criteria exist, but the most commonly used are
the single linkage or nearest-neighbor criterion [SIB 73], the
complete linkage or furthest-neighbor criterion [SOR 48] and
the average linkage criterion [SOK 58]. When the objects are
described by continuous variables, it is also possible to use
Ward's method.

1.2.2. The \( k \)-means algorithm

This section deals with the \( k \)-means algorithm, which is
the classical method in partitional clustering when the data
are a set of objects \( x_1, \ldots, x_n \) described by \( d \) continuous
variables. The objective of partitional (or non-hierarchical
clustering) is to define the partition of a set of objects into
clusters, that the objects in a cluster are more “similar” to
each other than to objects in other clusters. Starting from \( g \)
initial cluster centers, the \( k \)-means algorithm involves the
two following steps up to the convergence: assign each object
in \( \Omega \) to the nearest cluster center; use the centroids of the
different clusters as the new cluster centers. It can easily be
shown that this algorithm yields a stationary sequence of partition decreasing the within-group sum-of-squares criterion.

The term \( k \)-means actually covers a whole family of methods, and the algorithm previously described is only one example. Bock [BOC 07] has carried out an interesting survey of some historical issues related to the \( k \)-means algorithm. We can cite Dalenius [DAL 50, DAL 51], Lloyd's algorithm [LLO 57] in the context of scalar quantization in the one-dimensional case and Steinhaus [STE 56] for data in \( \mathbb{R}^d \) in the multidimensional case. Different strategies have been used: first, we have batch algorithms that process all the objects of the sample at each iteration, and incremental algorithms that process only one object at each iteration. Second, we have on-line or off-line training. In on-line training, each object is discarded after it has been processed (on-line training is always incremental). In off-line training, all the data are stored and can be accessed repeatedly (batch algorithms are always off-line). The term sequential is ambiguous, referring sometimes to incremental algorithms and sometimes to on-line learning. On-line \( k \)-means variants are particularly suitable when not all the data to be classified are available at the outset. The \( k \)-means algorithm previously described is an example of a batch algorithm. This batch version, which is the one used most often, was proposed by Forgy [FOR 65], Jancey [JAN 66] or Linde et al. [LIN 80] in vector quantization. The algorithm of MacQueen [MAC 67], who was the first to use the name “\( k \)-means”, and the neural gas algorithm [MAR 91b] are examples of on-line \( k \)-means.

If the aim is to find the partition minimizing the within-group sum-of-squares criterion, the \( k \)-means algorithm does not necessarily provide the best result, but simply a sequence of partitions, the value of whose criterion
will decrease, thus giving a local optimum. Since, in practice, convergence is reached very quickly (often in fewer than 10 iterations even with a large data set), the user can run $k$-means several times, with different random initializations and retain the best partition, i.e. that optimizes the criterion.

If the number of clusters is not known, several solutions are possible to solve this very difficult problem. For example, the best partition is sought for several numbers of classes and the number of classes are selected by choosing an elbow on the scree plot. It is also possible to add additional constraints relating, for example, to the number of objects by cluster or to the volume of a cluster (see, for instance, the Isodata algorithm [BAL 67]). Finally, there are other approaches using statistical methods, such as hypothesis tests or selection model criteria. This last approach, apparently the most interesting, consists of penalizing the criterion by a function depending on the number of classes, making the criterion “independent” of this number of classes. For instance, minimizing the within-group sum of squares can be viewed as maximizing a likelihood (see section 1.6), and selection model criteria such as the Akaike information criterion (AIC) or the Bayesian information criterion (BIC) can be used to determine the number of clusters.

Finally, it can be interesting to use $k$-means and Ward's method simultaneously: the two methods are similar in that they both attempt to minimize the within-group sum-of-squares criterion. This leads us to propose strategies using the two approaches, such as, for example, the hybrid method proposed by Wong [WON 82].

1.2.3. Other approaches

The dynamic cluster method proposed by Diday [DID 71, DID 76] is a generalization of the $k$-means
Co-Clustering

algorithm based on the quite powerful idea that the cluster centers are not necessarily centroids of clusters in \( \mathbb{R}^d \) and to replace them by centers that may take a variety of forms, depending on the problem to be solved. The \( k \)-medoids algorithm ([KAU 87]) is a typical dynamic cluster method where the cluster centers are objects of the set to cluster. Different versions have been proposed: partition around medoids (PAM) ([KAU 90]) and CLARANS ([NG 94]), which are more efficient for large volumes of data. This method is particularly well adapted when the data are given as a dissimilarity matrix \( d \). The \( k \)-modes algorithm [HUA 97, NAD 93] for categorical data is another example in which the centers are vectors of categories. A final example (adaptive distance, [DID 74, DID 77]), in which each center is a pair of point and distance, determines partition and distance simultaneously allowing the shapes of clusters to be taken into account.

Fuzzy clustering [RUS 69], developed to handle the notion of overlapped clusters, generalizes the classical approach in clustering by assuming that each element \( x_i \) can belong to more than one cluster with different levels. A fuzzy partition can therefore be represented by a fuzzy classification matrix \( c = \{ c_{ik} \} \) satisfying the following conditions: \( \forall i, k, c_{ik} \in [0, 1] \), \( \forall k, \sum_i c_{ik} > 0 \) and \( \forall i, \sum_k c_{ik} = 1 \). Bezdek [BEZ 81] proposed the fuzzy \( k \)-means algorithm, which can be viewed as a fuzzy version of \( k \)-means. The parameter estimation of a mixture model (see section 1.3) can also be viewed as a fuzzy clustering, and the associated EM algorithm is a more statistically formalized method that includes the notion of partial membership in classes. It has better convergence properties and is in general preferred to fuzzy \( k \)-means.

The self-organizing map (SOM) or Kohonen map [KOH 82] was first inspired by the adaptive formation of topology-conserving neural projection in the brain. Its aim is
to generate a mapping of a set of high-dimensional input signals onto a one- or two-dimensional array of formal neurons. Each neuron becomes representative of some input signals, such that the topological relationship among input signals in the input space is reflected, as faithfully as possible, in the arrangement of the corresponding neurons in the array (also called output space). When using this method for clustering, it is possible either to match each neuron with a unique cluster or to match many neurons to one cluster. In the latter case, the Kohonen algorithm produces a reduced representation of the original data set, and clustering algorithms may operate on this new representation. In the SOM literature, we refer to the clusters by the nodes or neurons, each of which has a weight in $\mathbb{R}^d$. The weights refer to the cluster means. The principal advantage of SOM is that it preserves the topology clustering. Generally, the neurons are arranged as a one- or two-dimensional rectangular grid preserving relations among the objects, also referred to as units. SOM is therefore a useful tool for visualizing clusters and evaluating their proximity in a reduced space. The Kohonen map can be viewed as an extension of the on-line $k$-means algorithm and, like $k$-means, requires the number of clusters (nodes of the grid) to be fixed and initial values to be selected. Different strategies can be used but initialization using principal component analysis (PCA) would appear to be an attractive and interesting approach.

High-dimensional data present a particular challenge to clustering algorithms. This is because of the so-called curse of dimensionality that leads to the sparsity of the data: in high-dimensional space, all pairs of points tend to be almost equidistant from one another. As a result, it is often unrealistic to define distance-based clusters in a meaningful way. Usually, clusters cannot be found in the original feature space because several features may be irrelevant for clustering, owing to correlation or redundancy. However,
clusters are usually embedded in lower dimensional subspaces, and different sets of features may be relevant for different sets of objects. Thus, objects can often be clustered differently in subspaces varying from the original feature space. Different approaches have been proposed. Subspace clustering seeks to find clusters in different subspaces within a data set. An example of this kind of approach is the CLIQUE algorithm [AGR 98]. It is a density-based method that can automatically find subspaces of the highest dimensionality such that high-density clusters exist in those subspaces. Subspace ranking aims at identifying all subspaces of a (high-dimensional) feature space that contain interesting clustering structures. The subspaces should be ranked according to this interestingness. Projected clustering is a method whereby the subsets of dimensions selected are specific to the clusters themselves. We can also cite the high-dimensional data clustering (HDDC) approach of Bouveyron [BOU 07]. In this approach, a family of Gaussian mixture models designed for high-dimensional data combining the ideas of subspace clustering and parsimonious modeling are used to develop a clustering method based on the EM algorithm.

Kernel clustering methods have attracted much attention in recent years [FIL 08]. They involve transforming a low-dimensional input space into a high-dimensional kernel-deduced feature space in which patterns are more likely to be linearly separable [SCH 02]. They have certain advantages when handling nonlinear separable data sets. We can also cite spectral clustering techniques that make use of the spectrum of the similarity matrix of the data to perform dimensionality reduction for clustering in fewer dimensions, and more generally, graph clustering techniques.

In this section, different approaches to clustering, essentially classical and generally based on numerical
criteria, have been reviewed. Unfortunately, defining these
criteria and using them successfully is not always easy. To
overcome these difficulties exist other approaches, such as
the mixture model approach, which is undoubtedly a very
useful contribution to clustering. It offers considerable
flexibility, gives a meaning to certain criteria and sometimes
leads to replacing criteria with new criteria with fewer
drawbacks. In addition, it provides solutions to the problem
of the number of clusters. The next section describes this
approach.

1.3. Model-based clustering and the mixture model

The clustering methods described in the previous two
sections are mainly heuristic techniques derived from
empirical methods, usually optimizing measurement criteria.
Implementing these solutions entails choosing not only a
metric reflecting the dissimilarity among the objects in the
set to be segmented, but also a criterion deriving from this
metric capable of measuring the degree of cohesion and
separation among classes. A rapid perusal of the lists of
metrics and criteria proposed in the clustering literature will
be enough to convince most readers that these are not easy
choices. A number of different probabilistic clustering
methods have been proposed, but the use of finite mixture
densities, which provides a sensible statistical model for the
clustering process, is now widespread.

Since their first use by Newcomb in 1886 for the detection
of outlier points, and then by Pearson in 1894 to identify two
separate populations of crabs, finite mixtures of distributions
have been employed to model a wide variety of random
phenomena. These models assume that measurements are
taken from a set of individuals, each of which belongs to one
of a number of different classes, while any individual’s
particular class is unknown. We might, for instance, know
the sizes of fish in a sample, but not their sex, which is
difficult to ascertain. Mixture models can thus address the
heterogeneity of a population, and are especially well suited
to the problem of clustering. This is an area where much
research has been done. McLachlan and Peel’s book [MCL 00]
is a highly detailed reference for this domain that has seen
considerable developments over the last few years. We will
first briefly recall the model and the problems of estimating
its parameters.

Finite mixture models, which assume that every class is
categorized by a probability distribution, are highly flexible
models that can take account of a variety of situations
including heterogeneous populations and outlier elements.
Because of the EM algorithm, which is particularly well
suited to this kind of context, a number of mixture models
have been developed in the field of statistics, and the use of
mixture models in clustering has been studied by authors
including Scott and Symons [SCO 71], Marriott [MAR 75],
Symons [SYM 81], McLachlan [MCL 82] and McLachlan and
Basford [MCL 88]. The mixture model approach is attractive
for several reasons. It corresponds to our intuitive idea of a
population composed of several classes, it is strongly linked to
reference methods such as the $k$-means algorithm and it is
able to handle a wide variety of special situations in a more
or less natural way. It is this approach that forms the subject
of this chapter.

In a finite probability mixture model, the data
$x = (x_1, \ldots, x_n)$ are taken to constitute a sample of $n$
independent instances of a random variable $X$ in $\mathbb{R}^d$. The
productivity density function (pdf) can be expressed as

$$f(x_i) = \sum_k \pi_k f_k(x_i), \quad \forall i \in I,$$
where \( g \) is the number of components, \( f_k \) are the pdf of each component and \( \pi_k \) are the mixture proportions (\( \pi_k \in [0,1] \forall k \) and \( \sum_k \pi_k = 1 \)). The principle of a mixture model is to suppose, given the proportions \( \pi_1, \ldots, \pi_g \) and the distributions \( f_k \) of each class, that the data are generated according to the following mechanism:

- \( z \): each individual is allotted to a class according to a categorical distribution with parameters \( \pi_1, \ldots, \pi_g \);
- \( x \): each \( x_i \) is assumed to arise from a random vector with pdf \( f_k \).

In addition, it is usually assumed that the components’ pdf \( f_k \) belong to a parametric family of pdf \( f(., \alpha) \). The pdf of the mixture can therefore be written as

\[
f(x_i; \theta) = \sum_k \pi_k f(x_i; \alpha_k), \quad \forall i \in I,
\]

where \( \theta = (\pi_1, \ldots, \pi_g, \alpha_1, \ldots, \alpha_g) \) is the parameter of the model. For example, the pdf of a mixture model for two univariate Gaussian distributions of variance 1 in \( \mathbb{R} \) is written as

\[
f(x_i; \pi, \mu_1, \mu_2) = \pi \varphi(x_i; \mu_1, 1) + (1 - \pi) \varphi(x_i; \mu_2, 1),
\]

where \( \varphi(.; \mu, \sigma^2) \) is the pdf of the univariate Gaussian distribution of mean \( \mu \) and variance \( \sigma^2 \). Figure 1.1 uses the pdf obtained from a mixture of three Gaussian components in \( \mathbb{R}^2 \) to illustrate this concept of a probability mixture.

Much effort has been devoted to the estimation of parameters for the mixture model, following the work of Pearson, whose use of the method of moments to estimate the five parameters \( (\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \pi) \) of a univariate Gaussian mixture model with two components required him to solve polynomial equations of degree nine. There have been a
number of studies [MCL 88, TIT 85] and different estimation methods have been envisaged. Apart from the method of moments, we also find graphic methods, the maximum likelihood method and Bayesian approaches. In this chapter, we will restrict ourselves to examining the maximum likelihood method using the EM algorithm, which is currently the most widely used. Before examining this method in section 1.4, we first draw the reader’s attention to certain difficulties which the estimation of parameters of a mixture model presents.

![Gaussian mixture in $\mathbb{R}^2$](image)

**Figure 1.1.** Gaussian mixture in $\mathbb{R}^2$

In certain situations, such as in the case of the crabs data previously described where the idea of the component has a precise physical basis, the number of components may be completely determined. Most often, however, the number of components is not known and must itself be estimated. It should be noted that if the number of components is taken to be an additional parameter, the mixture model may be seen as a semi-parametric compromise between a classical parametric estimation problem, when the number of components corresponds to a fixed constant, and a
non-parametric estimation problem, in this case via the kernel method, when the number of components is equal to the size of the sample. We assume from here onwards that number \( g \) of components is known, and later we will look at the proposed solutions for making this difficult choice.

If the problem is to be of any interest, the pdf of the mixture needs to be identifiable, which means that any two mixtures whose densities are the same must have the same parameters. A number of studies have addressed this problem and several difficulties arise. The first difficulty is due to the numbering of the classes. For example, in the case of a mixture with two components, the parameters 
\[(\pi_1, \pi_2), (\alpha_1, \alpha_2) \] and 
\[(\pi_2, \pi_1), (\alpha_2, \alpha_1)\], although different, obviously yield the same pdf: a mixture is consequently never identifiable. The difficulties to which this situation gives rise will depend on the estimation algorithms. In the case of the EM algorithm that we will use, it simply does not matter, however, this cannot be said of the Bayesian approach, where this situation is known as the “switching problem”. The second, considerably more awkward, difficulty may arise from the very nature of the component pdf. It may easily be established that a mixture of uniform or binomial distributions is not identifiable. Mixtures of Gaussian, exponential and Poisson distributions, however, are identifiable.

1.4. EM algorithm

Maximizing the log-likelihood of a mixture model

\[
L(\theta) = \log \left( \prod_i \sum_k \pi_k f(x_i, \alpha_k) \right)
\]

leads to likelihood equations that usually have no analytical solution. It may, nevertheless, be shown that if the parameter
\( \alpha_k \) is a vector of real numbers \( \alpha_{kr} \), the solution of these likelihood equations must satisfy

\[
\pi_k = \frac{1}{n} \sum_i \tilde{z}_{ik} \quad \forall k \quad \text{and} \quad \sum_i \tilde{z}_{ik} \frac{\partial \log f_k(x_i, \alpha_k)}{\partial \alpha_{kr}} = 0 \quad \forall k, r \quad [1.2]
\]

with

\[
\tilde{z}_{ik} = \frac{\pi_k f_k(x_i, \alpha_k)}{\sum_{k'} \pi_{k'} f_{k'}(x_i, \alpha_{k'})}. \quad [1.3]
\]

These equations suggest the following iterative algorithm:

1. start from an initial solution \( \theta \);
2. calculate the values \( \tilde{z}_{ik} \) from this parameter using equation [1.3];
3. update the parameter \( \theta \) on the basis of these values \( \tilde{z}_{ik} \) using equations [1.2];
4. continue from (2). If this algorithm converges, therefore, the fixed point obtained will satisfy the likelihood equations.

The procedure corresponds, in fact, to the application of the Dempster et al. [DEM 77] EM algorithm to the mixture model. Before describing this algorithm, we will define the concept of complete data on which it relies.

### 1.4.1. Complete data and complete-data likelihood

At the outset, we consider that the observed data \( x \) correspond to what is merely a partial knowledge of unknown data \( y \) that are termed complete data, the two being linked by a function \( x = T(y) \). The complete data might, for instance, be of the form \( y = (x, z) \), in which case \( z \) is known as missing information. This idea of complete data may either be meaningful for a model, which is the case for the mixture model, or it may be completely artificial. The likelihood \( f(y; \theta) \) calculated from these complete data is termed complete-data likelihood or, in the case of the mixture model, classification likelihood. Starting from the equation \( f(y; \theta) = f(y|x; \theta) f(x; \theta) \), we obtain the equation

\[
L(\theta) = L_{C}(\theta, z) - \log f(y|x; \theta) \quad [1.4]
\]
between the initial log-likelihood $L(\theta)$ and the complete-data log-likelihood $L_C(\theta, z)$.

### 1.4.2. Principle

The EM algorithm is based on the hypothesis that maximizing the complete-data likelihood is simple. Since this likelihood cannot be calculated – $y$ is unknown – an iterative procedure based on the conditional expectation of the log-likelihood for a value of the current parameter $\theta'$ is used as follows: first, calculating the conditional expectation for the two members of equation [1.4], we obtain the fundamental equation of the EM algorithm

$$L(\theta) = Q(\theta, \theta') - H(\theta, \theta'),$$

where $Q(\theta, \theta') = \mathbb{E}(L_C(\theta, z)|x, \theta')$ and $H(\theta, \theta') = \mathbb{E} \log f(y|x; \theta)|x, \theta')$.

Introducing the parameter $\theta'$ allows us to define an iterative algorithm to increase the likelihood. Using Jenssen's inequality, it can be shown that for fixed $\theta'$, the function $H(\theta, \theta')$ is the maximum for $\theta = \theta'$. The value $\theta$ that maximizes $Q(\theta, \theta')$, therefore, satisfies the equation

$$L(\theta) \geq L(\theta').$$

[1.5]

The EM algorithm involves constructing, from an initial solution $\theta^{(0)}$, the sequence $\theta^{(q)}$ satisfying $\theta^{(q+1)} = \arg \max Q(\theta, \theta^{(q)})$. Equation [1.5] shows that this sequence causes the criterion $L(\theta)$ to develop.
1.4.3. Application to mixture models

For the mixture model, the complete data are obtained by adding the original component $z_i$ to each individual member of the sample

$$y = (y_1, \ldots, y_n) = ((x_1, z_1), \ldots, (x_n, z_n)).$$

Coding $z_i = (z_{i1}, \ldots, z_{ig})$ where, let us recall, $z_{ik}$ equals 1 if $i$ belongs to component $k$ and 0 otherwise, we obtain the following equations

$$f(y; \theta) = \prod_i f(y_i; \theta) = \prod_i \sum_k \pi_k f(x_i; \alpha_k),$$

$$L_C(\theta, z) = \log(f(y; \theta)) = \sum_{i,k} z_{ik} \log(\pi_k f(x_i; \alpha_k)),$$

$$Q(\theta|\theta') = \sum_{i,k} E(z_{ik}|x, \theta') \log(\pi_k f(x_i; \alpha_k)).$$

Denoting as $\tilde{z}_{ik}$ the probabilities of belonging $E(z_{ik}|x, \theta') = P(z_{ik} = 1|x, \theta')$, the EM algorithm takes the following form:

– initialize: arbitrarily select an initial solution $\theta$;
– repeat the following two steps until convergence:
  - step E (expectation): calculate the probabilities of $x_i$ belonging to the classes, conditionally on the current parameter
    $$\tilde{z}_{ik} = \frac{\pi_k f(x_i; \alpha_k)}{\sum_{k'} \pi_{k'} f(x_i; \alpha_{k'})};$$
  - step M (maximization): maximize the log-likelihood conditionally on $\tilde{z}_{ik}$; the proportions are therefore obtained simply by the equation $\pi_k = \sum_i \tilde{z}_{ik}/n$, while the parameters $\alpha_k$ are obtained by solving the likelihood equations that depend on the mixture model employed.
1.4.4. Properties

Under certain conditions of regularity, it has been established that the EM algorithm always converges to a local likelihood maximum. It shows good practical behavior, but may, nevertheless, be quite slow in some situations. This is the case, for instance, when classes are very mixed. This algorithm, proposed by Dempster et al. in a seminal paper [DEM 77], often simple to implement, has gained widespread popularity and given rise to a large number of studies that are thoroughly covered in McLachlan and Krishnan's book [MCL 97].

1.4.5. EM: an alternating optimization algorithm

Hathaway [HAT 86] has shown that the EM algorithm applied to a mixture model may be interpreted as an alternating algorithm for optimizing a fuzzy clustering criterion. We make use of this fact below when we examine the links between estimating the parameters of a mixture model and fuzzy clustering. To obtain this result, Hathaway defines the criterion

\[ F_C(\tilde{z}, \theta) = L_C(\theta, \tilde{z}) + H(\tilde{z}), \quad [1.6] \]

where \( H(\tilde{z}) = -\sum_{i,k} \tilde{z}_{ik} \log \tilde{z}_{ik} \) is the entropy of the distribution \( \tilde{z} \). Moreover, if we denote as \( \tilde{z}_\theta \) the posterior distribution \( P(\tilde{z}|x, \theta) \), it can be shown, using equation [1.4], that the criterion \( F_C \) can also be expressed as

\[ F_C(\tilde{z}, \theta) = L(\theta) - KL(\tilde{z}, \tilde{z}_\theta), \quad [1.7] \]

where \( KL \) is the Kullback–Liebler divergence between two distributions.

The alternating algorithm for optimizing the criterion \( F_C \), therefore, becomes simple to implement:
minimizing for fixed $\theta$: equation [1.7] implies that $\tilde{z}$ must minimize $\text{KL}(\tilde{z}, \tilde{z}_\theta)$ and consequently $\tilde{z} = \tilde{z}_\theta$;

minimizing for fixed $\tilde{z}$: equation [1.6] shows that $\theta$ must maximize the expectation $L_C(\theta, \tilde{z})$.

Therefore, we are dealing with what are precisely the two steps of the EM algorithm. In addition, after each first step, we have $F_C(\tilde{z}, \theta) = F_C(\tilde{z}_\theta, \theta) = L(\theta)$, demonstrating that the EM algorithm increases the likelihood.

1.5. Clustering and the mixture model

1.5.1. The two approaches

Mixture models may be used in two different ways to obtain a partition of the initial data.

- The first, known as the mixture approach, estimates the parameters of the model and then determines the partition by allocating each individual to the class that maximizes the a posteriori probability $\tilde{z}_{ik}$ computed using these estimated parameters; this allocation is known as the maximum a posteriori probability (MAP) method.

- The second, the classification approach, was first presented by Scott and Symons [SCO 71] and developed further by Schroeder [SCH 76]; this approach involves creating a partition of the sample such that each class $k$ is made to correspond to a sub-sample respecting the distribution $f(., \alpha_k)$. This requires simultaneous estimation of the model parameters and the desired partition.

In this section, we describe the criterion that the latter approach optimizes, as well as the optimization algorithm usually employed in this situation. We then briefly compare the two approaches and examine links between these types of
methods and the more classical metrical approaches to clustering. We conclude the section by looking at how the mixture model may be interpreted in terms of fuzzy clustering.

1.5.2. Classification likelihood

Introducing the $z$ partition in the likelihood criterion is not an obvious step, and various ideas have been proposed. Scott and Symons [SCO 71] defined the criterion

$$L_{CR}(\theta, z) = \sum_k \sum_{i: z_{ik} = 1} \log f(x_i, \alpha_k)$$

in which the proportions do not appear. Symons [SYM 81], realizing that this criterion tends to yield classes of similar proportions, modified it so as to use the complete-data (or classification) log-likelihood described above

$$L_C(\theta, z) = \sum_k \sum_{i: z_{ik} = 1} \log \pi_k f(x_i, \alpha_k) = \sum_{i,k} z_{ik} \log \pi_k f(x_i, \alpha_k)$$

linked to the previous criterion by the equation

$$L_C(\theta, z) = L_{CR}(\theta, z) + \sum_k z_k \log \pi_k,$$

where $z_k$ is the cardinal of the class $k$. The quantity $\sum_k z_k \log \pi_k$ is a penalty term that disappears if all the proportions are made to be identical. The criterion $L_{CR}(\theta, z)$ can, therefore, be seen as a variant of classification likelihood, restricted to a mixture model where all classes have the same proportion.
1.5.3. **The CEM algorithm**

When seeking to maximize classification likelihood, it is possible to use a clustering version of the EM algorithm, obtained by adding a clustering step. This yields the very general clustering algorithm known as classification EM (CEM) [CEL 92], defined as follows:

– **step 0**: arbitrarily select an initial solution \( \theta \);

– **step E**: compute \( z_{ik} \) as in the EM algorithm;

– **step C**: obtain the \( z \) partition by allocating each \( x_i \) to the class that maximizes \( \tilde{z}_{ik} \) (MAP); this is equivalent to modifying the \( \tilde{z}_{ik} \) by replacing them with the nearest 1 or 0 values;

– **step M**: maximize the likelihood depending on the \( z_{ik} \); the estimations of the maximum likelihood among the \( \pi_k \) and the \( \alpha_k \) are obtained using the classes of the partition \( z \) as sub-samples, where the proportions are given by the formula \( \pi_k = \frac{1}{n} z_{ik} \), the \( \alpha_k \) being computed according to the particular mixture model selected.

Here, we have an alternating *dynamic cluster methods* [DID 79] type optimization algorithm, where the \( E \) and the \( C \) steps correspond to the allocation step, and the \( M \) step corresponds to the representation step.

It can be shown that this algorithm is stationary and that it increases the complete-data likelihood at each iteration, given some very general assumptions.

1.5.4. **Comparison of the two approaches**

The clustering approach, which determines the parameters at each iteration using truncated mixture model samples, yields a biased and inconsistent estimation, since
the number of parameters to be estimated increases as the size of the sample increases. Different authors have studied this problem and shown that it is usually preferable to use the mixture approach.

However, when the classes are well separated and membership relatively small, the clustering approach can sometimes give better results [CEL 93, GOV 96]. Moreover, the CEM algorithm is considerably faster than the EM algorithm, and it may be necessary to use it when computation time is limited, for example in real-time operations, or for very large volumes of data.

Finally, the clustering approach has the advantage of being able to present a large number of clustering algorithms as special cases of the CEM algorithm, which allows it to incorporate them into a probabilistic clustering approach. We will see in section 1.6.3, for example, that the \( k \)-means algorithm can be seen as a simple special case of the CEM algorithm. In particular, we will show that the optimized criteria, the within-group sum-of-squares criterion for continuous data and the information criterion for qualitative data correspond to the classification likelihood of a particular mixture model. These correspondences, studied in [GOV 89] and [GOV 90b], can be formalized by the following theorem.

**Theorem 1.1.**—If the clustering criterion can be expressed as

\[
W(z, \lambda, D) = \sum_{i,k} z_{ik} \Delta(x_i, \lambda_k),
\]

where \( z \) is a partition of the set to be segmented, \( \lambda = (\lambda_1, \ldots, \lambda_g) \) and \( \lambda_k \) are representatives of the class \( k \), and \( \Delta \) is a measure of the dissimilarity between an object \( x \) and the representative of a class, and if there exists a real \( r \) such that the quantity \( \int r^{-\Delta(x, \lambda)} dx \) is independent of \( \lambda \), this criterion is therefore equivalent to the classification
likelihood criterion of a mixture model with densities of the form \( f(x, \lambda) = \frac{1}{s} r^{-\Delta(x, \lambda)} \), \( s \) being a positive constant.

This theorem may be used equally well for continuous data as for discrete (binary or qualitative) data – either a Lebesgue measure or a discrete measure will be used accordingly. This theorem is important insofar as a great many clustering criteria can be put into this very general form; for example, it is the case for the intraclass inertia criterion, whose class representative is its center of gravity and where the distance \( D \) is the square of the Euclidean distance. It can also help us to fix the fields of application of these criteria and to suggest others.

1.5.5. Fuzzy clustering

In fuzzy clustering, it is no longer the case that an object either belongs or does not belong to a particular class. Instead, there are degrees of belonging. Formally, fuzzy clustering is characterized by a matrix \( c \) with terms \( c_{ik} \) satisfying \( c_{ik} \in [0, 1] \) and \( \sum_k c_{ik} = 1 \). Bezdek’s “fuzzy \( k \)-means” [BEZ 81], one of the most commonly encountered, involves minimizing the criterion

\[
W(c) = \sum_{i,k} c_{ik}^\gamma d^2(x_i, g_k),
\]

where \( \gamma > 1 \) is a coefficient for adjusting the degree of fuzziness, \( g_k \) is the center of the class and \( d \) is the Euclidean distance. It is required that \( \gamma \) be different from 1, otherwise the function \( W \) is minimal for values of \( c_{ik} = 0 \) or 1 and thus we have the usual within-group sum-of-squares criterion. The values usually recommended are between 1 and 2. Minimizing this criterion is achieved using an algorithm that alternates between the two following steps:
1) compute the centers: \( g_k = \frac{\sum_i c_{ik} x_i}{\sum_i c_{ik}} \);

2) compute the fuzzy partition: \( c_{ik} = \frac{C_i}{\|x_i - g_k\|^\gamma} \)

with \( C_i = \sum_{k'} \frac{1}{\|x_i - g_{k'}\|^\gamma} \).

Validating this kind of approach and, in particular, choosing the coefficient \( \gamma \) can be tricky. Therefore, estimating the parameters of a mixture model is an alternative, more natural, way of addressing this problem. The estimation of the \textit{a posteriori} probabilities \( \tilde{z}_{ik} \) of objects belonging to each class directly provides a fuzzy clustering, and the EM algorithm, applied to the mixture model, may be seen as a fuzzy clustering algorithm.

As mentioned above, Hathaway [HAT 86] went even further and showed that seeking to obtain a fuzzy partition and the parameter \( \theta \) using an optimization alternated with a fuzzy clustering criterion leads precisely to the two steps of the EM algorithm, which can therefore be considered as a fuzzy clustering algorithm. One may obtain the same result simply by applying the results from section 1.4.5 to the mixture model. Given that here the probability distribution \( \tilde{z} \) is defined by the vector \( (c_{ik}) \) and that we simply have \( \mathbb{E}_c(L_C(\theta, z)) = L_C(\theta, c) \), we show that the EM algorithm alternately maximizes the criterion

\[ W(c, \theta) = L_C(\theta, c) + H(c), \]

where \( L_C \) is the complete-data log-likelihood function where the partition \( z \) has been replaced by the fuzzy partition \( c \)

\[ L_C(\theta, c) = \sum_{i,k} c_{ik} \log (\pi_k f_k(x_i; \alpha)) \]
and $H$ is the entropy function

$$H(c) = -\sum_{i,k} c_{ik} \log c_{ik}.$$  

It is easy to show that if the entropy term of the criterion $W$ is removed, then, “hard” partitions are obtained at each step. The resulting algorithm is simply the CEM algorithm: the difference between the EM and CEM algorithms is the presence of the entropy term. If, when EM converges, the components are highly separated, the fuzzy partition $z(\theta)$ is close to a partition and we have

$$H(z(\theta)) \approx 0$$

and

$$L(\theta) = W(z(\theta), \theta) = L_C(\theta, z(\theta)) + H(z(\theta)) \approx L_C(\theta, z(\theta)).$$

1.6. Gaussian mixture model

We will now examine what happens to this approach when each class is modeled by a Gaussian distribution, which is a classical solution for continuous data.

1.6.1. The model

The pdf of the mixture can be written as

$$f(x; \theta) = \sum_k \pi_k \varphi(x; \mu_k, \Sigma_k),$$

where $\varphi$ is the pdf of the Gaussian multivariate distribution

$$\varphi(x; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2} (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) \right\}$$

and $\theta$ is the vector $(\pi_1, \ldots, \pi_g, \mu_1, \ldots, \mu_g, \Sigma_1, \ldots, \Sigma_g)$ formed by the proportions $\pi_k$ and the parameters $\mu_k$ and $\Sigma_k$, which
are, respectively, the mean vector and the covariance matrix of class $k$.

When the sample size is small, or when the dimension of the space is large, the number of parameters must be reduced so as to obtain more parsimonious models. To this end, the spectral decomposition of the matrices [BAN 93, CEL 95] may be used, allowing the covariance matrices to be parameterized uniquely as $\Sigma_k = \lambda_k D_k A_k D_k^t$, where the diagonal matrix $A_k$ with determinant 1 and decreasing values defines the shape of the class, the orthogonal matrix $D_k$ defines the direction of the class and the positive real number $\lambda_k$ represents the volume of the class. Thus, the mixture model is parameterized by the centers $\mu_1, \ldots, \mu_g$, the proportions $\pi_1, \ldots, \pi_g$, the volumes $\lambda_1, \ldots, \lambda_g$, the shapes $A_1, \ldots, A_g$ and the directions $D_1, \ldots, D_g$ of each class.

For example, when the data are in a plane, $D$ is a rotation matrix defined by an angle $\alpha$ and $A$ is a diagonal matrix with diagonal terms $a$ and $1/a$. Figure 1.2 shows the equidensity ellipse of this distribution depending on the values $\alpha$, $\lambda$ and $a$.

![Equidensity ellipse](image)

**Figure 1.2. Parameterization of a Gaussian class in the plane**

Using this parameterization, it becomes possible to propose solutions that can be seen as a middle way between, on the one hand, restrictive hypotheses (covariance matrices
proportional to the identity matrix, or covariance matrices identical for all classes) and, on the other hand, very general constraint-free hypotheses [BAN 93, CEL 95].

This parameterization also highlights two distinct notions that are often conflated under the rather vague heading of size: these are, first, the proportion of individuals present within a class and, second, the volume that a class occupies in space. It is quite possible for a class to have a small volume and a high proportion or, alternatively, a large volume but a low proportion.

We will now look at what happens to the CEM algorithm and the classification likelihood criterion in the case of the Gaussian mixture model. It should be noted that a similar approach could be applied to the EM algorithm.

1.6.2. **CEM algorithm**

1.6.2.1. **Clustering step**

Each \( x_i \) is allocated to the class that maximizes the probability of membership \( \tilde{z}_{ik} = \frac{\pi_k \varphi(x_i; \mu_k, \Sigma_k)}{(\sum_{k'} \pi_{k'} \varphi(x_i; \mu_{k'}, \Sigma_{k'}))} \), that is to say \( \pi_k \varphi(x_i; \mu_k, \Sigma_k) \) or, equivalently, the class that minimizes \(-\log(\pi_k \varphi(x_i; \mu_k, \Sigma_k))\), which can be written as

\[
\begin{align*}
\mathbf{d}^2_{\Sigma_k^{-1}}(x_i, \mu_k) + \log |\Sigma_k| - 2 \log \pi_k,
\end{align*}
\]

where \( \mathbf{d}^2_{\Sigma_k^{-1}}(x_i, \mu_k) \) is the quadratic distance \((x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)\).
1.6.2.2. **Step M**

Here, for a given partition $z$, we have to determine the parameter $\theta$ that maximizes $L_C(\theta, z)$, which is equal (ignoring one constant) to

$$-\frac{1}{2} \sum_k \left( \sum_i z_{ik} (x_i - \mu_k)^\top \Sigma_k^{-1} (x_i - \mu_k) + z_k \log |\Sigma_k| - 2 z_k \log \pi_k \right).$$

The parameter $\mu_k$ is thus necessarily the center of gravity $\bar{x}_k = \frac{1}{z_k} \sum_i z_{ik} x_i$ and the proportions, if they are not constrained, satisfy $\pi_k = z_k/n$. The parameters $\Sigma_k$ must then minimize the function

$$F(\Sigma_1, \ldots, \Sigma_g) = \sum_k z_k \left( \text{trace}(S_k \Sigma_k^{-1}) + \log |\Sigma_k| \right), \quad [1.9]$$

where

$$S_k = \frac{1}{z_k} \sum_i z_{ik} (x_i - \bar{x}_k)(x_i - \bar{x}_k)^\top$$

is the covariance matrix of the class $k$. We now examine three particular situations.

1.6.3. **Spherical form, identical proportions and volumes**

We now look at the most straightforward situation where all classes have a Gaussian spherical distribution with the same volume and the same proportion. The covariance matrices are written as $\Sigma_k = \lambda D_k I_d D_k^\top = \lambda I_d \forall k$, and formula [1.8] shows that individuals can be allotted to the different classes simply by using the usual Euclidean distance $d^2(x_i, \mu_k)$. Function $F$, therefore, becomes

$$F(\lambda) = \frac{1}{\lambda} \sum_k z_k \text{trace}(S_k) + nd \log \lambda = \frac{1}{\lambda} n \text{trace}(S_W) + nd \log \lambda,$$
where $S_W = \frac{1}{n} \sum_k z_k S_k$ is the within-group covariance matrix, thus giving us $\lambda = \frac{\text{trace}(S_W)}{d}$. The classification likelihood is written as

$$L_C(\theta, z) = -\frac{nd}{2} \log \text{trace}(S_W) + \text{cst.}$$

Maximizing the classification likelihood is therefore equivalent to minimizing the within-group sum-of-squares criterion $\text{trace}(S_W)$. Moreover, the CEM algorithm is simply the $k$-means algorithm. This means that to use the within-group sum-of-squares criterion is to assume that classes are spherical and have the same proportion and the same volume.

1.6.4. Spherical form, identical proportions but differing volumes

We now take the model described above and modify it slightly to include classes with different volumes. The covariance matrices are now written $\Sigma_k = \lambda_k I_p$, and formula [1.8] shows that individuals are allotted to classes according to the distance

$$\frac{1}{\lambda_k} D^2(x_i, \mu_k) + d \log \lambda_k.$$

The distance from a point to the center of a class has been modified by an amount that depends on the volume of the class. This modification has important repercussions; for example, the regions of separation, which in the previous case were hyperplanes, become hyperspheres. It may be shown that the minimized criterion can be written as

$$\sum_k \log \text{trace}(S_k).$$
With this model, we can very easily recognize situations such as the situation shown in Figure 1.3. Here, the two classes have been simulated with two spherical Gaussian distributions which have the same proportions but widely differing volumes. The result obtained using the classical intraclass inertia criterion corresponds to a separation of the population by a straight line and therefore bears no relation at all to the simulated partition. With the variable-volume model, the obtained partition, shown by the circle, is very close to the initial clustering.

\[ \Sigma_k = \Sigma \]

It will be noted that without the help of the mixture model, it would have been difficult, on the basis of a simple metrical interpretation, to come up with the distance and the criterion used in this approach.

1.6.5. Identical covariance matrices and proportions

Our final example is where all classes have the same form and the same proportion. The covariance matrix of each class can thus be written as \( \Sigma_k = \Sigma \). It can be shown that individuals are now allotted to classes on the basis of the distance \( d_{\Sigma}^2(x_i, \mu_k) \) and that the criterion to be minimized
may be written as $|S_W|$, which serves to justify the use of this criterion, sometimes proposed in a metrical context [FRI 67], without reference to the Gaussian model.

### 1.7. Binary data

We now turn, still within a broad discussion of clustering methods based on probability distribution mixture models, to the clustering of sets of individuals measured using binary variables.

#### 1.7.1. Binary mixture model

As the Gaussian model is often chosen to model each component of the mixture when variables are continuous, the log-linear model [AGR 90, BOC 86] is a natural choice when variables are binary. The complete or saturated log-linear model, where each class has a multinomial distribution with $2^d$ values, is not really applicable in the case of a mixture. Instead, we use a log-linear model with sufficient constraints. The simplest example is the independence model that assumes that, conditionally on membership of a class, the binary variables are independent. From this, we obtain the latent class model [GOO 74, LAZ 68], which we will now examine.

In the binary case, each $x_{ij}$ has a Bernoulli distribution whose probability distribution takes the form

$$f(x_{ij}; \alpha_{kj}) = (\alpha_{kj})^{x_{ij}}(1 - \alpha_{kj})^{1-x_{ij}} \text{ where } \alpha_{kj} = P(x_{ij} = 1|k).$$

Therefore, the distribution of the class $k$ is

$$f_k(x_i; \alpha_k) = \prod_j (\alpha_{kj})^{x_{ij}}(1 - \alpha_{kj})^{1-x_{ij}} \text{ where } \alpha_k = (\alpha_{k1}, \ldots, \alpha_{kp}),$$
and the mixture model chosen considers that the data \( x_1, \ldots, x_n \) constitute a sample of independent instances from a random \( \{0, 1\}^d \) probability vector

\[
f(x_i; \theta) = \sum_k \pi_k f(x_i; \theta_k) = \sum_k \pi_k \prod_j (\alpha_{kj})^{x_{ij}} (1 - \alpha_{kj})^{1-x_{ij}},
\]

where the parameter \( \theta \) is constituted by the proportions \( \pi_1, \ldots, \pi_g \) and by the parameter vector \( \alpha = (\alpha_1, \ldots, \alpha_g) \) of each component.

The problem that arises is how to estimate these parameters, and possibly the origin class as well. As in the case of the Gaussian model, the estimation may be obtained using the EM or the CEM algorithms described above. The only differences concern the computation of the parameters \( \alpha_{kj} \) that becomes \( \alpha_{kj} = \frac{\sum_i z_{ik} x_{ij}^j}{\sum_i z_{ik}} \) for the first, and \( \alpha_{kj} = \frac{\sum_i z_{ik} x_{ij}^j}{\sum_i z_{ik}} = \% \) of 1 for the second. Intensive comparisons between the two algorithms EM and CEM were performed by Govaert and Nadif [GOV 96].

1.7.2. Parsimonious model

The number of parameters of this latent class model is equal to \((g - 1) + g \times d\), where \( g \), it will be recalled, is the number of classes and \( d \) is the number of binary variables. In the case of the complete log-linear model, however, the number of parameters is equal to \( 2^d \). For example, when \( g = 5 \) and \( d = 10 \), the number of parameters for the two models is, respectively, 54 and 1,024. Given that one of the identifiability conditions of the model is that the number of states is greater than the number of parameters, there is a clear interest in being able to propose even more
parsimonious models. To this end, the model may be restated as follows
\[
f(x; \theta) = \sum_k \pi_k \prod_j (\varepsilon_{kj})^{x_{ij} - a_{kj}}(1 - \varepsilon_{kj})^{1 - |x_{ij} - a_{kj}|},
\]
where \( a_{kj} = 0, \varepsilon_{kj} = \alpha_{kj} \) if \( \alpha_{kj} < 0.5 \)
\( a_{kj} = 1, \varepsilon_{kj} = 1 - \alpha_{kj} \) otherwise.

The parameter \( \alpha_k \) is, thus, replaced by the two following parameters:

- a binary vector \( a_k \) representing the center of the class and which is the most frequent binary value for each variable;
- a vector \( \varepsilon_k \) belonging to the set \( [0, 1/2]^d \) that defines the dispersion of the component, and represents the probability of any particular variable's having a value different from that of the center.

We are, thus, led to the parameters used by Aitchinson and Aitken [AIT 76] for discrimination with non-parametric estimation via the kernel method. Starting from this formulation, we arrive at parsimonious situations by stipulating certain constraints: the \([\varepsilon]\) model is defined by stipulating that the dispersion should not depend either on the component or on the variable, the \([\varepsilon_k]\) model by stipulating that it should depend only on the component, and the \([\varepsilon^j]\) model by stipulating that it should depend only on the variable.

For example, in the simplest case, the \([\varepsilon]\) model, given identical proportions \( (\pi_k = 1/g) \), the clustering approach results in the complete-data log-likelihood being maximized
\[
L_C(\theta, z) = \log \frac{\varepsilon}{1 - \varepsilon} \sum_{i,k} z_{ik} d(x_i, a_k) + nd \log(1 - \varepsilon),
\]
which is to say that the criterion

$$W(z, \theta) = \sum_{i,k} z_{ik} d(x_i, a_k)$$

where

$$d(x_i, a_k) = \sum_j |x_{ij} - a_{kj}|$$

is minimized.

Step $E$ of the CEM algorithm, therefore, consists simply of allotting each individual to the class $k$ that minimizes $d(x_i, a_k)$. At step $M$, the parameters $a_{kj}$ for each variable $j$ correspond to the majority binary values in each class $k$. A class, therefore, corresponds to a binary vector, and the criterion is easy to interpret: it is simply the number of differences among individuals and their representative in the partition $z$. To use this binary clustering criterion proposed by different authors [GOW 74, GOV 90a] is therefore to assume that the data come from a particular latent class model.

1.7.3. Examples of application

To illustrate this approach, we have taken the Stouffer-Toby data set [STO 51], analyzed within the framework of the latent class model by Goodman [GOO 74]. Table 1.1 contains the reactions, classed as one of two possible attitudes, shown by 216 subjects placed in four different conflict situations. We compare the latent class model (here requiring nine parameters) with the log-linear model, with an interaction of order 2, and a very similar number of parameters (11). It is interesting to note that for these data the deviance drops from 7.11 in the case of the linear model to a value of 2.72 in the case of the latent class model. The parameters obtained for the latent class model are shown in Table 1.2.
### Table 1.1. Stouffer-Toby data

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>42</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>38</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
</tbody>
</table>

### Table 1.2. Results obtained by EM for the latent class model

<table>
<thead>
<tr>
<th></th>
<th>$p_k$</th>
<th>$a_{k1}$</th>
<th>$a_{k2}$</th>
<th>$a_{k3}$</th>
<th>$a_{k4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.279</td>
<td>0.993</td>
<td>0.940</td>
<td>0.927</td>
<td>0.769</td>
</tr>
<tr>
<td>2</td>
<td>0.721</td>
<td>0.714</td>
<td>0.330</td>
<td>0.354</td>
<td>0.132</td>
</tr>
</tbody>
</table>

#### 1.8. Categorical variables

We now extend the results of the previous section to categorical data.

#### 1.8.1. Multinomial mixture model

As for binary data, we will examine the latent class model and therefore assume that the $d$ qualitative variables are independent, conditionally on their membership to a class. If
\( \alpha_{kj}^h \) is the probability that the \( j \)th variable takes the modality \( h \) when an individual belongs to the class \( k \), therefore the pdf of the mixture can be written as

\[
f(x_i; \theta) = \sum_k \pi_k f(x_i; \alpha_k) = \sum_k \pi_k \prod_j \prod_{h=1}^{m_j} (\alpha_{kj}^h)^{x_{ij}} \]

where the parameter \( \theta \) is defined by the proportions \( \pi_1, \ldots, \pi_g \) and by the parameters \( \alpha_k = (\alpha_{kj}^h; j = 1, \ldots, d; h = 1, \ldots, m_j) \) of the pdf of each component.

As before, estimating the parameter \( \theta \), and possibly estimating the native class of each of the \( x_i \), may be achieved by maximizing the likelihood \( L(\theta; x) \) using the EM algorithm, or by maximizing the complete-data likelihood \( L_C(\theta, z) \) using the CEM algorithm. In the case of the EM algorithm, the computation of the parameters \( \alpha_k \) at step M is defined by the equation

\[
\alpha_{kj}^h = \frac{\sum_i z_{ik} x_{ij}^h}{\sum_i z_{ik}} \]

where \( z_{ik} \) are the probabilities obtained in the usual fashion at step E. In the case of the CEM algorithm, the computation of the parameters \( \alpha_k \) becomes

\[
\alpha_{kj}^h = \frac{\sum_i z_{ik} x_{ij}^h}{\sum z_k} \]

where \( z_k \) is the partition obtained by the MAP from the probabilities \( \bar{z}_{ik} \).

We now look at what happens to the complete-data likelihood criterion when the clustering approach is used with the assumption that the proportions \( \pi_k \) are constant. If we denote

\[
s_k^h = \sum_i z_{ik} x_{ij}^h, s_j^h = \sum_i x_{ij}^h, s_k = \sum_j \sum_{h=1}^{m_j} s_k^h \quad \text{and} \quad s^h = \sum_i \sum_{h=1}^{m_j} x_{ij}^h = nd, \]

we can easily show that the equation

\[
L_C(z, \theta) = \sum_{k,j} \sum_{h=1}^{m_j} s_k^h \log \alpha_{kj}^h
\]

is obtained.
Given that, at convergence, \( a_{h_{kj}}^k = s_{ij}^k / z_{kj} \), it can be shown [CEL 91] that the CEM algorithm maximizes the information criterion [BEN 73a]

\[
H(z, J) = \sum_{k,j} \sum_{h=1}^{m_j} s_{ij}^k \log \frac{s_{ij}^k}{s_j^k s_i^h}
\]

which represents the information from the initial table, retained by the partition \( z \), and yielding results very close to the \( \chi^2 \) criterion

\[
\chi^2(z, J) = \sum_{k,j} \sum_{h=1}^{m_j} \frac{(s_{ij}^k - s_j^k s_i^h)^2}{s_j^k s_i^h}.
\]

Therefore, it follows that to seek a partition into \( g \) classes maximizing the information criterion or the \( \chi^2 \) criterion (approximately equivalent) is to assume that the data derive from a latent class model.

A parallel may be drawn here with the analysis of multiple correspondences. It is not difficult to see that with the geometrical representation used in this factorial analysis, the \( \chi^2 \) criterion is quite simply the familiar criterion of intraclass inertia.

### 1.8.2. Parsimonious model

The number of parameters \((g - 1) + g \times \sum_j (m_j - 1)\) required by the latent class model that we have just described is usually considerably smaller than the number of parameters \( \prod_j m_j \) required by the complete log-linear model. For example, for a number of classes \( g \) is equal to 5 and a number of qualitative variables \( d \) are equal to 10, and where the number of modalities \( m_j \) is 4 for all the variables, the number of parameters for the two models is, respectively, 154
and $10^6$. In many cases, this number will be quite excessive, and more parsimonious models are called for.

To this end, we begin by remarking that if, for each variable $j$, the modality of highest probability is denoted as $h^*$, the model may therefore be re-parameterized as follows

$$f(x_i; \theta) = \sum_k \pi_k \prod_j \left( (1 - \varepsilon_{kj}^*)^{1 - d(x_{ij}; a_{kj})} \prod_{h \neq h^*} (\varepsilon_{kj}^h | x_{ij} - a_{kj}^h) \right),$$

where $a_{kj} = (a_{1kj}, \ldots, a_{mkj})$ with $a_{kj}^h = 1$ if $h = h^*$ and 0 otherwise, $\varepsilon_{kj} = (\varepsilon_{1kj}, \ldots, \varepsilon_{mkj})$ where $\varepsilon_{kj}^h = 1 - \alpha_{kj}^h$ if $h = h^*$ and $\alpha_{kj}^h$ otherwise, and $\delta(x_{ij}, a_{kj}) = 0$ if $x_{ij}$ and $a_{kj}$ take the same modality and 1 otherwise. Like for binary data, the vector $a_k = (a_{k1}, \ldots, a_{kd})$ may be interpreted as the center of the class $k$ and the vectors $\varepsilon_k^j$ as dispersions. For example, if the parameter $\alpha_{kj}$ is equal to the vector $(0.7, 0.2, 0.1)$, the new parameters become $a_{kj} = (1, 0, 0)$ and $\varepsilon_{kj} = (0.3, 0.2, 0.1)$.

With the model restated in this way, it is possible to introduce simple constraints, such as requiring non-majority modalities to have the same dispersion

$$\begin{cases} 
\varepsilon_{kj}^h = \varepsilon_{kj} \\
\varepsilon_{kj}^h = (1 - \varepsilon_{kj})/(m_j - 1) \text{ for } h \neq h^*.
\end{cases}$$

This gives us a model used in discrimination, where the number of parameters has been reduced from $(g - 1) + g \times \sum_j (m_j - 1)$ to $(g - 1) + \sum_j (m_j - 1)$.

Even more parsimonious models may be obtained if additional constraints are placed on dispersions, for example by requiring that $\varepsilon_{kj}^h$ should not depend on the variable (model $[\varepsilon_k]$), on the class (model $[\varepsilon_j]$) or on neither the variable nor the class (model $[\varepsilon]$). If we restrict ourselves to this last model $[\varepsilon]$, and if we require proportions to be equal,
the complete-data log-likelihood can be expressed quite simply

\[ L_C(z, \theta) = \log \frac{\varepsilon}{1 - \varepsilon} \sum_k \sum_{i=1}^n z_{ik} d(x_i, a_k) + nd \log(1 - \varepsilon), \]

where \( d(x_i, a_k) \) is a distance reflecting the number of different modalities between the vector \( x_i \) and the center \( a_k \). At the clustering step of the CEM algorithm, the individuals \( i \) are thus allotted to the class \( k \) that minimizes \( d(x_i, a_k) \), and at step M, the co-ordinates \( a_{kj} \) of the centers \( a_k \) are obtained by taking the majority of modalities. Furthermore, Jollois and Nadif [JOL 02] considered the clustering of categorical data under the classification maximum likelihood approach. In this setting, with a parsimonious multinomial mixture model, they defined a generalization of the \( k \)-modes criterion [HUA 98]. They showed that \( k \)-modes is just a particular version of CEM and the \( k \)-modes criterion is associated with a multinomial mixture model \([\varepsilon]\) with supplementary constraints that are too restrictive: the proportions are assumed to be equal and the variables to have the same number of categories. They conducted experiments showing the superiority of CEM with the model on \( k \)-modes when these assumptions are not verified.

In practice, we suggest taking very simple models by class, for example latent class models with a single parameter, and then increasing the number of components if necessary.

When the qualitative variables are ordinal, it is possible either to convert the data into binary data or to use an approach similar to the approach we have just described, taking into account the order that exists among the modalities.
1.9. Contingency tables

To measure the information provided by a contingency table, we need to evaluate the links existing between the two sets $I$ and $J$ as discussed in the Introduction. Several measures of association exist, and one of the most frequently employed is the phi-squared criterion $\phi^2$ (see Introduction and Chapter 4). This criterion, used, for example, in correspondence analysis (CA), is defined as follows

$$\phi^2(I, J) = \sum_{i,j} \frac{(p_{ij} - p_i p_j)^2}{p_i p_j}.$$ 

The phi-squared criterion can be used to evaluate the quality of a partition $z$ of $I$: to this end, we associate the partition $z$ with the phi-squared $\phi^2(z, J)$ of the contingency table with $g$ rows and $d$ columns obtained from the initial table in computing the sum of the rows of each cluster. It can be shown that

$$\phi^2(I, J) \geq \phi^2(z, J) \quad [1.10]$$

and therefore the proposed regrouping necessarily leads to a loss of information. The objective of classification is to find the partition $z$ that minimizes this loss, i.e. which maximizes $\phi^2(z, J)$. We notice that when the row profiles are equal for each cluster, inequality [1.10] becomes $\phi^2(z, J) = \phi^2(I, J)$, and in this particular case, there is no loss of information. In addition, the problem is meaningful only when the number of clusters is fixed. Otherwise, the optimal partition is simply the partition where each element of $I$ forms a cluster.

1.9.1. MNDKI2 algorithm

The MNDKI2 algorithm is based on the same geometrical representation of a contingency table as that used in CA. This
representation is justified for several reasons, in particular, because of the similar role played by each of the two dimensions in the analyzed table, and also because of the property of distributional equivalence, which implies stable results when agglomerating elements with similar profiles. In this representation, each row \( i \) corresponds to a point vector \( \mathbb{R}^d \) defined by the profile \( p'_j \) weighted by the marginal frequency \( p_k \). The distances among profiles is not defined by the usual Euclidean metric, but instead by the weighted Euclidean metric, known as the \textit{chi-squared metric} \( D^2 \), defined by the diagonal matrix \( \text{diag}(\frac{1}{p_1}, \ldots, \frac{1}{p_d}) \).

If \( z \) is a partition of the rows, we can define the frequencies 
\[ p_{kj} = \sum_i z_{ik} p_{ij} \]
and the average row profile of the \( k \)th cluster
\[ p^k_j = \left( \frac{p_{k1}}{p_k}, \ldots, \frac{p_{kd}}{p_k} \right)^t, \]
where \( p_{kj} = \sum_j p_{kj} \). With this representation, we can show after some calculation that the total of squared distances \( T \), the between-cluster sums of squares \( B(z) \) and the within-cluster sums of squares \( W(z) \) can be written as
\[ T = \sum_i p_i D^2(p^i_j, p_j) = \phi^2(I, J), \]
\[ B(z) = \sum_k p_k D^2(p^k_j, p_j) = \phi^2(z, J), \]
and
\[ W(z) = \sum_{i,k} z_{ik} p_i D^2(p^i_j, p^k_j). \]

The traditional equation between the total of squared distances, the within-cluster sums of squares and the
between-cluster sums of squares $T = W(z) + B(z)$ leads to the following equation

$$
\phi^2(I, J) = W(z) + \phi^2(z, J).
$$

The term $W(z)$, therefore, represents the information lost when grouping the elements according to the partition $z$, and $\phi^2(z, J)$ corresponds to the information which is preserved. Consequently, since the quantity $\phi^2(I, J)$ does not depend on the partition $z$, looking for the partition maximizing the criterion $\phi^2(z, J)$ is equivalent to looking for the partition minimizing criterion $W(z)$. To minimize this criterion, it is possible to apply $k$-means to the set of profiles with the $\chi^2$ metric. An iterative algorithm, known as MNDKI2, is thus obtained, locally maximizing $\phi^2(z, J)$.

The question that naturally arises is: which probabilistic model does the criterion $\phi^2(z, J)$ minimized by the MNDKI2 algorithm correspond to? The answer to this question will not only shed some light on this criterion, but it will also help us to propose other criteria. This is a question that we will focus on. Unfortunately, unlike the standard $k$-means algorithm, the MNDKI2 algorithm does not correspond to the classification approach associated with a mixture model [GOV 89]. However, using a mixture of multinomial distributions, examined in the following section, we will obtain approximately similar properties.

1.9.2. Model-based approach

1.9.2.1. Multinomial mixture

A contingency table can be obtained using a mixture of multinomial distributions by the following process of simulation [GOV 07]:
– z: each individual is allotted to a class according to a multinomial distribution with parameters \((\pi_1, \ldots, \pi_g)\);

– \(x_I = (x_1, \ldots, x_n)\): generate each row sum \(x_i\) according to a discrete distribution \(\psi\) such as a Poisson or a binomial distribution;

– x: each \(x_i\) is assumed to arise from a multinomial distribution with parameters \(x_i\) and \(\alpha_{k1}, \ldots, \alpha_{kd}\).

Thus, if \(\theta = (\pi_1, \ldots, \pi_g, \alpha_{11}, \ldots, \alpha_{gd})\) denotes the parameter of the model and \(\varphi\) is the multinomial distribution of the \(k\)th component, the pdf of this model is written as

\[
f(x_i; \theta) = \psi(x_i) \sum_k \pi_k \varphi(x_i; x_i, \alpha_{k1}, \ldots, \alpha_{kd})
\]

\[
= \psi(x_i) \sum_k \pi_k \frac{x_{i1}! \ldots x_{is}! \alpha_{x_{i1}k1} \ldots \alpha_{x_{is}kd}}{x_i!}
\]

\[
= A \sum_k \pi_k \alpha_{x_{i1}k1} \ldots \alpha_{x_{is}kd},
\]

where \(A = \psi(x_i) \frac{x_{i1}! \ldots x_{is}!}{x_i!}\) does not depend on the parameter \(\theta\).

The log-likelihood (without the additional constant \(\log A\)) can therefore be written as

\[
L(\theta; x) = \sum_i \log \sum_k \pi_k \alpha_{x_{i1}k1} \ldots \alpha_{x_{is}kd}, \tag{1.11}
\]

and the complete data log-likelihood is as follows

\[
L(\theta; x, z) = \sum_{i,k} z_{ik} \left( \ln \pi_k + \sum_j x_{ij} \log \alpha_{kj} \right). \tag{1.12}
\]

The classical problem is, therefore, to estimate the parameter \(\theta\) from the sample. In the clustering context, the
mixture model serves to find the component from which each row arises. Next, we see how the EM and CEM algorithms allow us to achieve this goal.

1.9.2.2. EM algorithm

For this multinomial mixture model, the application of the EM algorithm described in section 1.4.3 to the sample \( x = (x_1, \ldots, x_n) \) leads in the M-step to \( \alpha_{kj} = \frac{\sum \tilde{z}_{ik} x_{ij}}{\sum \tilde{z}_{ik}} \). The different steps of EM are then expressed in algorithm 1.1.

**Algorithm 1.1 Multinomial EM**

**input:** \( x, \ g \)

**initialization:** \( z, \ \pi_k = \frac{z_k}{n}, \ \alpha_{kj} = \frac{\sum \tilde{z}_{ik} x_{ij}}{\sum \tilde{z}_{ik}}. \)

**repeat**

**E-step.** \( \tilde{z}_{ik} \propto \pi_k \alpha_{k1} x_{i1} \cdots \alpha_{kd} x_{id} \)

**M-step.** \( \pi_k = \frac{\tilde{z}_k}{n}, \ \alpha_{kj} = \frac{\sum \tilde{z}_{ik} x_{ij}}{\sum \tilde{z}_{ik}}. \)

**until** convergence

**return** \( \pi, \alpha \)

In the maximum likelihood approach of the classical mixture model, after we have estimated the parameter \( \theta \), we can give a probabilistic clustering of the \( n \) rows in terms of their fitted posterior probabilities of component membership, and obtain a partition using a classification step that assigns each object to the component of the mixture to which it has the highest posterior probability of belonging.

1.9.2.3. CEM algorithm

Recall that in this classification approach, a C-step that converts the posterior probabilities \( \tilde{z}_{iks} \) to a discrete classification is included prior to performing the M-step. The different steps of the CEM algorithm are then expressed in algorithm 1.2.
Algorithm 1.2 Multinomial CEM

input: \( x, g \)
initialization: \( z, \pi_k = \frac{z_k}{n}, \alpha_{kj} = \frac{x_{ik} x_{ij}}{\sum_i z_{ik} x_{ij}} \)

repeat
  E-step. \( \tilde{z}_{ik} \propto \pi_k \alpha_{kj} x_{ij} \)
  C-step. \( z_i = \arg\max_k \tilde{z}_{ik} \)
  M-step. \( \pi_k = \frac{z_k}{n}, \alpha_{kj} = \frac{x_{ik} x_{ij}}{\sum_i z_{ik} x_{ij}} \)
until convergence
return \( \pi, \alpha, z, \)

Having established an estimate of the parameters, and denoting \( p_{kj} = \frac{\pi_{kj}}{x_k} \), we can express the criterion as

\[
L(\theta; x, z) = \sum_k z_k \ln \pi_k + \sum_{k,j} x_{kj} \log \frac{x_{kj}}{x_k} = \sum_k z_k \ln \pi_k + n \sum_{k,j} p_{kj} \log \frac{p_{kj}}{p_k p_j} + n \sum_j p_j \log p_j. \tag{1.13}
\]

Note that the term \( \sum_{k,j} p_{kj} \log \frac{p_{kj}}{p_k p_j} \) is the mutual information \( I(z, J) \) quantifying the information shared between \( z \) and \( J \). This can easily be shown using the definition in terms of entropies

\[
I(z, J) = H(z) + H(J) - H(z, J),
\]

where \( H(\cdot) \) is the entropy. This mutual information can be linked to the \( \phi^2 \) criterion as follows: first, using the equalities \( \sum_{k,j} p_k p_j = 1 \) and \( \sum_{k,j} p_{kj} = 1 \), we have the equation

\[
\sum_{k,j} \left( \frac{p_{kj} - p_k p_j}{p_k p_j} \right)^2 = \sum_{k,j} p_k p_j \left( \frac{p_{kj}}{p_k p_j} - 1 \right)^2.
\]
Second, using the approximation $x^2 - 1 \approx 2x \log x$, excellent in the neighborhood of 1 and good in the interval $[0, 3]$, the approximation
\[
\sum_{k,j} p_{k,p,j} \left( \left( \frac{p_{kj}}{p_{k,p,j}} \right)^2 - 1 \right) \approx 2 \sum_{k,j} p_{kj} \log \frac{p_{kj}}{p_{k,p,j}}
\]
can be obtained [BEN 73b]. Finally, we have the following approximation
\[
\sum_{k,j} p_{kj} \log \frac{p_{kj}}{p_{k,p,j}} \approx \frac{1}{2} \sum_{k,j} \left( p_{kj} - p_{k,p,j} \right)^2,
\]
and therefore,
\[
I(z, J) \approx \frac{1}{2} \phi^2(z, J).
\]

Therefore, from equations [1.13] and [1.14], when the proportions are fixed, the maximization of $L_C(\theta; z)$ is equivalent to the maximization of the mutual information $I(z, J)$, and approximately equivalent to the maximization of the phi-squared criterion $\phi^2(z, J)$: the use of the two criteria $\phi^2(z, J)$ and $I(z, J)$, therefore, is based on the implicit assumption that the data arise from a mixture of multinomial distributions.

1.9.3. Illustration

To illustrate the results obtained by MNDKI2, we will use a CA representation. Let us recall that CA is an exploratory multivariate technique that converts a contingency table into a particular type of graphical display in which the rows and the columns of the matrix are depicted as points [BEN 73b, GRE 88b, LEB 84]. It can be used on any two-way table, sparse or not, as the case may be. It projects the rows and columns of a data matrix into points within a graph in a
Co-Clustering

Euclidean space. The graph is, therefore, used to gain some understanding of the data and to extract information from it. To show the links between MNDKI2 and CA, we will use a comparison of time-budgets as an example [JAM 76]. We have a data matrix where \( x_{ij} \) represents the amount of time spent on a variety of activities \( i \) by a population \( j \) during a given time period. The set \( I \) comprises 10 activity clusters: prof (professional), tran (transport), home (housework), child (activities pertaining to childcare), shop (shopping), wash (washing and personal care), meal (mealtime), sleep, tv (television) and leis (other leisure activities). The set \( J \) is composed of 28 types of population characterized by gender, country, professional activity and marital status. The vector of letters identifying each population can be interpreted as follows: m or w (man or woman), a or na (active or not active professionally), s or ns (single or not single), us, we, ea or yu (USA, western country, eastern country or Yugoslavia); for instance, mnsyu corresponds to a man, not single and from Yugoslavia. We have presented the data in Table 1.3.

Here, we present the best result obtained by MNDKI2 from among 10 random initial positions when the number of clusters in partition \( z \) is 3. The initial \( \phi^2(I, J) \) value is 9658.38 and the resulting \( \phi^2(z, J) \) value is 8386.83. The percentage of \( \phi^2 \) accounted for by the partition is very good in this small example: more than 86% of the \( \phi^2 \) is preserved. The clusters in the obtained partition \( z \) are the following: cluster 1: home, child; cluster 2: prof, tran; and cluster 3: sleep, wash, leis, meal, shop, tv.

The column profiles \( \frac{p_{ij}}{p_i p_j} \) (with a multiple coefficient \( f_i \)) reorganized according to \( z \) are reported in Table 1.4. We observe the similarity of the profiles belonging to each cluster. The most interesting values are those that are a long way from the mean 1. They characterize the partition: for example, the category \( \text{wnaus} \) is a characteristic of the clusters 1 and 2.
Table 1.3. Transposed time-budget data matrix

<table>
<thead>
<tr>
<th></th>
<th>prof tran home child shop wash meal sleep tv leis</th>
</tr>
</thead>
<tbody>
<tr>
<td>maus</td>
<td>610 140 60 10 120 95 115 760 175 315</td>
</tr>
<tr>
<td>waus</td>
<td>475 90 250 30 140 120 100 775 115 365</td>
</tr>
<tr>
<td>wnaus</td>
<td>10 0 495 110 170 110 130 785 160 430</td>
</tr>
<tr>
<td>mnsus</td>
<td>615 141 65 10 115 90 115 765 180 305</td>
</tr>
<tr>
<td>wnsus</td>
<td>179 29 421 87 161 112 119 776 143 373</td>
</tr>
<tr>
<td>msus</td>
<td>585 115 50 0 150 105 100 760 150 385</td>
</tr>
<tr>
<td>waus</td>
<td>482 94 196 18 141 130 96 775 132 336</td>
</tr>
<tr>
<td>mawe</td>
<td>652 100 95 7 57 85 150 807 115 330</td>
</tr>
<tr>
<td>wawe</td>
<td>510 70 307 30 80 95 142 815 87 262</td>
</tr>
<tr>
<td>wnaue</td>
<td>20 7 567 87 112 90 180 842 125 367</td>
</tr>
<tr>
<td>mswe</td>
<td>655 97 97 10 52 85 152 807 122 320</td>
</tr>
<tr>
<td>wnswe</td>
<td>168 22 529 69 102 83 174 825 119 392</td>
</tr>
<tr>
<td>mswe</td>
<td>642 105 72 0 62 77 140 812 100 387</td>
</tr>
<tr>
<td>wswe</td>
<td>389 34 262 14 92 97 147 848 84 392</td>
</tr>
<tr>
<td>mayu</td>
<td>650 140 120 15 85 90 105 760 70 365</td>
</tr>
<tr>
<td>wayu</td>
<td>560 105 375 45 90 90 95 745 60 235</td>
</tr>
<tr>
<td>wnyu</td>
<td>10 10 710 55 145 85 130 815 60 380</td>
</tr>
<tr>
<td>nswy</td>
<td>650 145 112 15 85 90 105 760 80 357</td>
</tr>
<tr>
<td>wnsy</td>
<td>260 52 576 59 116 85 117 775 65 295</td>
</tr>
<tr>
<td>mayu</td>
<td>615 125 95 0 115 90 85 760 40 475</td>
</tr>
<tr>
<td>wse</td>
<td>430 39 318 23 112 96 102 774 45 409</td>
</tr>
<tr>
<td>msea</td>
<td>650 142 122 22 76 94 100 764 96 334</td>
</tr>
<tr>
<td>wsea</td>
<td>578 106 338 42 106 94 52 752 64 228</td>
</tr>
<tr>
<td>mnsea</td>
<td>652 133 134 22 68 94 102 762 122 310</td>
</tr>
<tr>
<td>wsea</td>
<td>434 77 431 60 117 88 105 770 73 229</td>
</tr>
<tr>
<td>msea</td>
<td>627 148 68 0 88 92 86 770 58 463</td>
</tr>
<tr>
<td>wsea</td>
<td>433 86 296 21 128 102 94 758 58 379</td>
</tr>
</tbody>
</table>

To illustrate the relationship between CA and MNDKI2, we have shown in Figure 1.4 the representation of I on the first two axes that account for 84% of $\phi^2$. We can observe that clusters 1 and 2 are strongly opposed and cluster 3 is the middle cluster.

1.10. Implementation

There are a number of software developments implementing the methods described in this chapter, not least
among them is the MIXMOD\(^1\) program. In this section, we give a quick overview of the problems that software implementations need to address.

### Table 1.4. The \( \frac{p_{ij}}{p_i} \times \frac{p_{ij}}{p_j} \) profiles reorganized according to the partition \( z \)

<table>
<thead>
<tr>
<th></th>
<th>prof tran</th>
<th>home child</th>
<th>shop wash meal sleep</th>
<th>tv leis</th>
</tr>
</thead>
<tbody>
<tr>
<td>maus</td>
<td>1359 1624</td>
<td>216 300</td>
<td>1103 1000 985 968 1758 903</td>
<td></td>
</tr>
<tr>
<td>waus</td>
<td>1058 1044</td>
<td>901 899</td>
<td>1286 1263 856 987 1155 874</td>
<td></td>
</tr>
<tr>
<td>wnaus</td>
<td>22 0</td>
<td>1785 3297</td>
<td>1562 1158 1113 1000 1607 1232</td>
<td></td>
</tr>
<tr>
<td>mnsus</td>
<td>1370 1635</td>
<td>234 300</td>
<td>1056 947 984 974 1807 874</td>
<td></td>
</tr>
<tr>
<td>wnsus</td>
<td>399 336</td>
<td>1518 2607</td>
<td>1479 1179 1019 988 1436 1069</td>
<td></td>
</tr>
<tr>
<td>maus</td>
<td>1304 1334</td>
<td>180 0</td>
<td>1378 1105 856 968 1507 1103</td>
<td></td>
</tr>
<tr>
<td>wsus</td>
<td>1074 1091</td>
<td>707 539</td>
<td>1296 1369 822 987 1326 963</td>
<td></td>
</tr>
<tr>
<td>mawe</td>
<td>1454 1161</td>
<td>343 210</td>
<td>524 896 1001 1217 1039 875 752</td>
<td></td>
</tr>
<tr>
<td>wawe</td>
<td>1137 813</td>
<td>1108 900</td>
<td>736 1001 1217 1039 875 752</td>
<td></td>
</tr>
<tr>
<td>wnasw</td>
<td>45 81</td>
<td>2047 2611</td>
<td>1030 949 1543 1074 1257 1053</td>
<td></td>
</tr>
<tr>
<td>mnswe</td>
<td>1461 1127</td>
<td>350 300</td>
<td>478 896 1303 1029 1227 918</td>
<td></td>
</tr>
<tr>
<td>wnswe</td>
<td>362 247</td>
<td>1844 1999</td>
<td>906 845 1440 1015 1155 1086</td>
<td></td>
</tr>
<tr>
<td>mswe</td>
<td>1432 1220</td>
<td>260 0</td>
<td>570 812 1200 1035 1006 1111</td>
<td></td>
</tr>
<tr>
<td>wawe</td>
<td>882 401</td>
<td>961 427</td>
<td>860 1039 1280 1099 858 1143</td>
<td></td>
</tr>
<tr>
<td>mayu</td>
<td>1448 1624</td>
<td>433 450</td>
<td>781 947 899 968 703 1046</td>
<td></td>
</tr>
<tr>
<td>wayu</td>
<td>1248 1218</td>
<td>1352 1349</td>
<td>827 947 813 949 603 674</td>
<td></td>
</tr>
<tr>
<td>wnyu</td>
<td>22 116</td>
<td>2560 1648</td>
<td>1332 895 1113 1038 605 1089</td>
<td></td>
</tr>
<tr>
<td>mnsyu</td>
<td>1449 1683</td>
<td>404 450</td>
<td>781 948 899 968 804 1024</td>
<td></td>
</tr>
<tr>
<td>wnsyu</td>
<td>579 603</td>
<td>2077 1768</td>
<td>1066 895 1002 987 653 845</td>
<td></td>
</tr>
<tr>
<td>msyu</td>
<td>1370 1450</td>
<td>343 0</td>
<td>1057 947 728 968 402 1361</td>
<td></td>
</tr>
<tr>
<td>wayu</td>
<td>928 1041</td>
<td>1156 695</td>
<td>1037 1019 880 994 456 1182</td>
<td></td>
</tr>
<tr>
<td>waeu</td>
<td>1448 1648</td>
<td>440 659</td>
<td>698 990 856 973 964 957</td>
<td></td>
</tr>
<tr>
<td>mnea</td>
<td>53 93</td>
<td>2142 2158</td>
<td>1452 969 1096 1070 864 1141</td>
<td></td>
</tr>
<tr>
<td>mnsea</td>
<td>1454 1544</td>
<td>483 660</td>
<td>625 990 874 971 1226 889</td>
<td></td>
</tr>
<tr>
<td>wnsea</td>
<td>974 899</td>
<td>1564 1810</td>
<td>1082 933 905 987 738 661</td>
<td></td>
</tr>
<tr>
<td>mnsu</td>
<td>1397 1717</td>
<td>245 0</td>
<td>809 969 736 981 583 1327</td>
<td></td>
</tr>
<tr>
<td>wsea</td>
<td>983 1017</td>
<td>1088 641</td>
<td>1199 1094 820 984 594 1107</td>
<td></td>
</tr>
</tbody>
</table>

1 www.mixmod.org.
1.10.1. Choice of model and of the number of classes

Clustering methods are often justified heuristically, and choosing the "right" method or the "right" number of classes can be a problem that is difficult, and often badly stated. The use of clustering methods based on mixture models allows us to place the problem within the more general framework of the selection of probabilistic models.

In the Bayesian context, choosing the most probable model calls for frequently used selection criteria such as Schwarz’s [SCH 78] BIC criterion comprising two terms: the first is likelihood, which tends to favor the more complex model, and the second is a penalizing term, an increasing function of the number of the model’s parameters. Worth mentioning is the ICL criterion [BIE 00] which, taking the objective of the clustering into account, generally provides good solutions.

1.10.2. Strategies for use

Maximizing the likelihood criterion via the EM algorithm or maximizing the clustering likelihood via the CEM algorithm always involves obtaining a series of solutions that see the criterion increase to a local maximum, and which are
therefore dependent on the initial position selected by the algorithm. The strategy usually adopted for obtaining a "good" solution is to run the algorithm several times from different starting points and to retain the best solution. For example, see [BIE 03], where some subtle and effective strategies are examined, including an initial phase in which the algorithm is run a large number of times without waiting for complete convergence.

1.10.3. Extension to particular situations

We have seen that the mixture model in clustering can cope with a variety of situations (spherical or non-spherical classes, equal or unequal proportions, etc.) and deal with both continuous and binary data. In this section, we briefly list some clustering problems that the mixture model approach addresses quite naturally, illustrating its adaptability to particular situations.

Noisy data: atypical or outlier data (measurement errors, etc.) generally perturb clustering methods quite considerably. Getting mixture models to take account of noise can be a simple matter, for example by adding a uniformly distributed class or by using distributions less sensitive to atypical elements, such as Laplace distributions.

Incomplete labeling in discrimination: in discrimination we often have, in addition to the learning sample whose class is known, a (sometimes large) set of observations whose class is not known. Making use of these unlabeled observations, which can significantly improve the results of the discrimination, can be easily accomplished by introducing observations whose membership to a class is not brought into question during the iterations of the algorithm to the EM and CEM algorithms.
Spatial data: the mixture model is based on the hypothesis that the vector \( z = (z_1, \ldots, z_n) \) grouping the classes of the different observations is an independent sample. There are, however, more complex situations, such as the segmentation of pixels in image processing, where this hypothesis must be rejected. In these cases, the mixture model may be extended to the clustering of geographically localized multivariate observations such as hidden Markov fields, so as to include this type of data.

Block clustering: the clustering methods described thus far were all designed to classify individuals, or occasionally variables, but there are other methods, often known as block or simultaneous clustering methods, which process the two sets simultaneously and organize the data into homogeneous blocks. Here too, it is possible to extend the use of mixture models [GOV 02] by using a latent block model generalizing the mixture model [GOV 03, GOV 05, GOV 06, GOV 08]. In the following chapters, we will focus on this model.

1.11. Conclusion

In this chapter, we have attempted to show the advantages of using mixture models in clustering. This approach provides a general framework capable of taking into account specificities in the data and in the problem. Moreover, a probabilistic model means being able to harness the entire set of statistical results in proposing solutions to difficult problems such as the choice of the model or the number of classes.

Obviously, one of the difficulties with this approach is in deciding whether the selected mixture model is realistic for the data in question. However, as Everitt [EVE 93] has rightly observed, it is not a difficulty specific to this approach. We cannot avoid choosing a method's underlying hypotheses simply by “concealing” them.