A regularized approach of instances x variables co-clustering for exploratory data analysis

Aichetou Bouchareb

THÈSE DE DOCTORAT

Université Paris 1 panthéon-Sorbonne

École doctorale : Sciences Mathématiques de Paris Centre (ED 386)

présentée par

Aichetou Bouchareb

Pour obtenir le grade de

Docteur de l'Université Paris 1 panthéon-Sorbonne

Spécialité Mathématiques Appliquées

A regularized approach of instances × variables co-clustering for exploratory data analysis

Soutenue le 28 Novembre 2018 devant le jury composé de :

M.	Julien Jacques	PR	Université de Lyon - Lyon 2	(Rapporteur)
M.	Mohamed Nadif	PR	Université Paris Descartes	(Rapporteur)
M.	Gilbert Saporta	PR émérite	Conservatoire National des Arts et Métiers	(Examinateur)
M.	Gilles Bisson	CR CNRS	Laboratoire d'Informatique de Grenoble	(Examinateur)
M.	Fabrice Clérot	Chercheur senior	Orange Labs	(Examinateur)
M.	Fabrice Rossi	PR	Université Paris 1 Panthéon-Sorbonne	(Directeur)
M.	Marc Boullé	Chercheur senior	Orange Labs	(Co-encadrant)

Abstract

Co-clustering is a class of unsupervised data analysis techniques aiming at extracting the underlying dependency structure between the rows and columns of a data table in the form of homogeneous blocks, known as co-clusters. These techniques can be distinguished into those that aim at simultaneously clustering the instances and variables, and those that aim at clustering the values of two or more variables of a data set. Most of these techniques are limited to variables of the same type, and are hardly scalable to large data sets while providing easily interpretable clusters and co-clusters.

Among the existing value based co-clustering approaches, MODL is suitable for processing large data sets with several numerical or categorical variables. In this thesis, we propose a value based approach, inspired by MODL, to perform a simultaneous clustering of the instances and variables of a data set with potentially mixed-type variables.

The proposed co-clustering model provides a Maximum A Posteriori based summary of the data that can be used as it is for exploratory analysis of the data. When the summary is large, exploratory analysis tools, such as model coarsening, can be used to simplify the co-clustering which facilitates the interpretation of the results. We show that the proposed co-clustering approach can handle large data and extract easily interpretable clusters from mixed data with more than 10 millions observations. We also show the robustness of the approach, its capacity to extract inter-dependence between the variables, and its good behavior in extreme cases such as in the case of pattern-less data and in the case of perfectly correlated variables.

Résumé

Le co-clustering est une classe de techniques d'analyse non supervisée visant à extraire la structure sous-jacente de dépendance entre les lignes et les colonnes d'un tableau de données sous la forme de blocs homogènes, appelés co-clusters. Ces techniques peuvent être différenciées en deux types: celles qui effectuent un groupement simultané des instances et des variables d'une matrice de données, et celles qui effectuent un groupement des valeurs de deux ou plusieurs variables. Toutefois, la plupart de ces méthodes se limitent à des variables du même type et sont difficilement adaptables à des bases de données de grande taille, tout en fournissant des clusters facilement interprétables.

Parmi les méthodes basées sur la classification des valeurs, MODL permet de traiter des données de grande taille et de réaliser une partition de plusieurs variables, numériques et/ou catégorielles. Dans cette thèse, nous proposons une approche de classification croisée, inspirée de MODL et basée sur le groupement des valeurs, pour effectuer un groupement simultané des instances et des variables d'un ensemble de données contenant des variables potentiellement de type mixte.

Le modèle proposé est basé sur l'estimation par Maximum A Posteriori et fournit un résumé de la base de données, exploitable pour l'analyse exploratoire. Lorsque ce résumé est très grand, des outils d'analyse exploratoire, comme la fusion successive des clusters, peuvent être utilisés pour simplifier le co-clustering, ce qui facilite l'interprétation des résultats. Nous montrons que l'approche proposée permet de traiter des données volumineuses et d'extraire des clusters facilement interprétables à partir de données mixtes comportant plus de 10 millions d'observations. Nous montrons également la robustesse de l'approche, sa capacité à extraire l'interdépendance entre les variables, et son bon comportement dans des cas extrêmes comme dans le cas des données sans motifs et dans le cas des variables parfaitement corrélées.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, the amounts of data collected from different sources, in various formats, and for various application areas is growing not only in the number of objects and attributes, but also in the complexity of the patterns to be extracted from the data. This has led to an increasing need for the development of techniques and tools to assist the analyst in extracting useful information (knowledge), from the rapidly growing volumes of data. The development of such techniques would enable intelligent and automatic analysis, exploration and organization of large and complex data, to extract and understand information. Moreover, with the emergence of connected objects, the volumes of available data and their diversity can only keep growing.

On the one hand, the increasing amount of data and the development of data mining techniques makes knowledge discovery tasks especially interesting for large companies, since they allow the data to be considered as a useful resource in the decision-making process. On the other hand, the increasing complexity of the data creates several challenges for the researchers, provided that many existing techniques are not appropriate to analyze large complex data. For example, consider the field of market analysis, where millions of transactions are observed. Data analysis techniques are used to analyze and summarize the information contained within these large data sets and to draw conclusions about the data and the studied objects. The collected data can be anything from demographic descriptors of the customers (age, gender, social class, occupation, education, income) to markers of customer preferences such as ratings. The analysis tools can range from simple analysis tools such as graphical displays such as bar charts and histograms, two-way tables such as contingency tables, and quantile plots (Amant and Cohen 1995) to more complex tools such as applying machine learning techniques (Everitt et al. 2011) to learn a buying pattern or to create a grouping of customers. Finally, the extracted conclusions can be used in recommendation systems.

However, the growing size of data sets has led to an increasing demand for efficient and noise tolerant data summarizing techniques for data compression and analysis, while respecting constraints in terms of memory usage and computation time.

1.1 Problem definition and main objectives

The main theme of this thesis is the use of co-clustering in exploratory analysis. Given a data matrix where the rows represent objects and columns represent their features, the goal of a co-clustering technique is to *simultaneously* extract clusters of objects and clusters of features. The co-clustering techniques try to exploit the interdependence between the objects and their descriptive features to create groups of objects and groups of features or of feature values in a way that best expresses the level of association between these groups. Hence expressing the association between the two sets.

With mixed-type commercial data in mind (such as e-commerce and consumer/product data), we seek a simultaneous clustering technique that would provide an easily exploitable summary of the data. When working with such data, a first level of complexity arises from the properties of the various attributes used to describe each data object, such as the fact that they are categorical or numerical, the cardinality of the domains, and the dependencies that may exist between different attributes. A second level of complexity arises from the fact that these data are considerably large and may have missing values. A third level of complexity arises from the variety of tasks that can be performed to analyze the data and from the existence of several alternative ways to perform each task. Thus, depending on the nature of knowledge one wants to extract, some techniques are more suitable than others.

Our goal is to propose a co-clustering method that handles data with mixed-type attributes, handles missing data, provides easily interpretable clusters and, overall, a good summary of the data and an evaluation measure.

1.2 Outline of the thesis

This manuscript is organized as follows.

- In Chapter 2, we define the problem of co-clustering and explore the existing literature on the subject. In this chapter, we will notice the multitude of solutions, their main differences, advantages and limitations.

- In Chapter 3, we propose a new approach for co-clustering mixed-type data. The approach is based on a user defined pre-processing step followed by a value oriented co-clustering technique. In this chapter, we show that the proposed approach enables extracting easily interpretable clusters of objects, captures local as well as global dependencies between the variables, and that it scales to data sets containing tens of thousands of objects and hundreds of thousands of entries.

- Chapter 4 describes a co-clustering model that formalizes the approach proposed in Chapter 3. The model eliminates the pre-processing phase by simultaneously inferring an optimized partitioning of each variable

and performing a co-clustering, by optimizing a Maximum A Posteriori (MAP) based model selection criterion. The model requires no user parameter and it enables associating values coming from different variables by setting the data matrix entries as the statistical units.

- Chapter 5 provides experimental results on synthetic and real-world data sets. These experiments highlight the main features of the co-clustering model proposed in Chapter 4, some of the possible ways in which a co-clustering model can be exploited, and provide a didactic explanation of how the model works and how it differs from the solution proposed in Chapter 3.

- Finally, in Chapter 6 we draw concluding remarks and highlight possible future perspectives and use cases for the proposed co-clustering model.

1.3 Publications

The work presented in this thesis is the subject of the following publications.

1. The work presented in Chapter 3 has contributed to the paper:
   Bouchareb, A., Boullé, M., Clérot, F., and Rossi, F. (2017a). Application du coclustering à l'analyse exploratoire d'une table de données. In *17ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2017*, volume RNTI-E-33, pages 177–188.

2. An extended version of Bouchareb et al. (2017a) is the subject of:
   Bouchareb, A., Boullé, M., Clérot, F., and Rossi, F. (2018a). Co-clustering based exploratory analysis of mixed-type data tables. In Pinaud, B., Gandon, F., Bisson, G., and Guillet, F., editors, *Accepted for publication in Advances in Knowledge Discovery and Management Vol. 8 (AKDM-8)*. Springer.

3. The work presented in Chapter 4 has contributed to the paper:
   Bouchareb, A., Boullé, M., Rossi, F., and Clérot, F. (2018c). Un modèle bayésien de co-clustering de données mixtes. In *Extraction et Gestion des Connaissances, EGC 2018, Paris, France, January 23-26, 2018*, pages 275–280.

4. To perform a co-clustering of mixed data, we have extended the Latent Block Models to the case of data containing binary and numerical variables. This extension is not presented in the core of this thesis, but is the subject of:
   Bouchareb, A., Boullé, M., and Rossi, F. (2017b). Co-clustering de données mixtes à base des modèles de mélange. In *17ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2017, 24-27 Janvier 2017, Grenoble, France*, pages 141–152.

5. Appendix B contains an extended version of the work published in Bouchareb et al. (2017b). This extended version is the subject of:
Bouchareb, A., Boullé, M., Clérot, F., and Rossi, F. (2018b). Model based co-clustering of mixed numerical and binary data. In Pinaud, B., Gandon, F., Bisson, G., and Guillet, F., editors, *Accepted for publication in Advances in Knowledge Discovery and Management Vol. 8 (AKDM-8)*. Springer.

Part I

State of the art

Chapter 2

Co-clustering for exploratory data analysis

This chapter gives an introduction to the subject of exploratory data analysis. In particular, we focus on the co-clustering based techniques. The objective is to lay out the motivating backgrounds for the rest of the thesis by introducing the most commonly known co-clustering techniques, their advantages and drawbacks as well as their domains of application.

This chapter is organized as follows. First, we briefly outline the context of exploratory analysis in Section 2.1. Section 2.2 presents the types of data sets we wish to analyze. In Section 2.3, we give a brief overview of discretization methods and their use for homogenizing the data as well as some of the most commonly used multivariate data description techniques, namely clustering and dimensionality reduction techniques. In particular, we illustrate the various choices involved in the process of data clustering, starting from the choice of the variables to the choice of the number of clusters, and introduce dimensionality reduction as a commonly used technique for data visualization and for association extraction. Section 2.4 introduces co-clustering as an extension of the clustering problem, the most commonly used co-clustering techniques, the types of structures they extract and their domains of applicability. In particular, we will distinguish between row and column based approaches (Section 2.5) and value based approaches (Section 2.6). Section 2.7 summarizes the main limitations of these methods and sets the motivation for the next chapter.

2.1 Introduction

The amount of data available nowadays is too large to process manually. Hence, one of the most common activities of the data analyst consists in trying to extract some *essence* information from an abundance of data. This urge for information extraction is particularly present in the industrial context where *data* is abundant and the necessity to exploit *information* becomes pressing. For example, companies like Orange hold information about their clients, their purchase history, and their preferences. With a large num-

ber of clients and ultimately a large number of attributes, exploratory data analysis becomes an essential tool in decision making. In particular, there is a need to use the collected data about users and their preferences (such as age, the products purchased, ratings, ...) to create reliable recommendations for a set of clients of interest or make general marketing policies like offering discounts on the products generally purchased together.

Data analysis tools differ in their objectives and underlying hypothesis. For example, these techniques are usually divided into two main types: supervised and unsupervised methods. In the supervised learning approach, also called predictive analysis, the goal is to *learn* a mapping from a set of inputs x to a set of outputs y, given a so called training set of labeled input-output pairs $D = \{(x_i, y_i)\}_{i=1}^{I}$, where I is the size of the training set (Murphy 2012). These techniques are called predictive because, ultimately, the goal is to predict, using the learned mapping, the output value y_j for new unlabeled values x_j. Examples of supervised methods include: neural networks, simple and multiple linear regression, and support vector machines (see: Maimon and Rokach (2010), Bishop (2006), Schölkopf and Smola (2002)).

The second main type of data analysis techniques is the unsupervised learning approach where only a set of unlabeled inputs $D = \{x_i\}_{i=1}^{I}$ is given, and the goal is to find *interesting structure in the data* (which is sometimes called knowledge discovery (Murphy 2012). The work presented here falls in this context of information extraction and knowledge discovery in databases. As Maimon and Rokach (2010) put it:

Knowledge Discovery in Databases (KDD) is an automatic, exploratory analysis and modeling of large data repositories. KDD is the organized process of identifying valid, novel, useful, and understandable patterns from large and complex data sets.

Therefore, the goal is to find *interesting patterns* by organizing and summarizing the data in a way that extracts *humanly understandable conclusions*. However, the term "interesting" is not well defined in the sense that the types of expected patterns are either unknown or task dependent, and that there is no natural error metric as in the case of predictive analysis, for example. These features make descriptive analysis a more complex task since the problem is not well defined and lacks a universal evaluation measure. On the other hand, these same features make descriptive analysis a flexible task and data dependent tool as there are less performance constraints and the analyst simply uses a set of discovery tools in order to *make statements* about the data and *describe* what it is or what it shows, through simplification. Instead of computing an error metric, the questions to be asked in an exploratory analysis context are of the type: have the right tools been chosen with respect to the task at hand, have the tools been used correctly, how credible are the statements and conclusions.

In the following, we focus on exploratory analysis techniques, which are a

subset of unsupervised methods, to lay out the motivational backgrounds for the work presented in this thesis, which falls in the framework of extracting relevant patterns from the data.

The process of performing an exploratory analysis encompasses a wide variety of choices that need to be made in order to extract any knowledge from data. The extracted knowledge, if any, is the composite result of all the choices. The first of these choices is to define what can be considered as "data", both in the objective sense and with respect to the type of knowledge we want to extract. The second choice concerns the definition of the exploratory analysis tools and the type of patterns one is seeking.

2.2 Data representation

Generally, the *data* $\mathcal{D}$ can be of any type and form. However, for easy manipulation of raw data, the data to be analyzed is usually represented as a set of points (called objects or instances) in a K-dimensional space of features (called variables). That is, data are generally represented in a rectangular table with I rows for the set of instances and K columns corresponding to the variables.

Choosing the instances and variables to analyze is a crucial phase and has an important influence on the results. This choice has to take into account the aim of the study. In particular, the variables have to describe the phenomenon being analyzed (Anderberg 1973). For example, in census data, the instances are individual persons and the features could be any thing from *age*, *income*, *number of family members*, *marital status*, ..., to *level of education* and *household electricity consumption*. In market analysis, when modeling purchasing patterns, the data can consist of a binary matrix where each column represents a product, each row represents a client, and an entry is equal to 1 if the corresponding product was purchased by the corresponding client.

Regardless of the application domain, let $\mathbf{X} = (x_{ij})_{1 \leq i \leq I, 1 \leq j \leq K}$ be the matrix representation of the data containing the observed data values where I is the number of instances, K is the number of variables, and the entry $x_{ij} = v$ says that the value of the j^{th} variable is v for the i^{th} instance.

Depending on the measured feature, a variable can be:

- categorical (also referred to as qualitative) to designate a variable for which the space of values is a finite set of un-ordered values,
- numerical (also referred to as quantitative) to designate a variable for which the values can be ranked in order and are subject to meaningful arithmetic operations.

Depending on the nature of the variables, data can be uni-type or mixed. Uni-type data is numerical when all variables are numerical and categorical when all variables are categorical (note that binary variables are a special

case of the categorical type but with only two categories). We refer to mixed data when both types of variables are present.

The goal of this thesis is to introduce an exploratory analysis technique for *large and mixed data sets*. More precisely, our goal is to analyze data containing millions of values (up to 10 millions) while providing easily interpretable results. But first, in the coming sections, we introduce the most commonly used exploratory analysis techniques. The literature shows that these techniques are mostly adapted for uni-type data and their performance can be limited when the data is large and complex in the sense that they would extract global summaries about the instances or variables but can miss subtle patterns like local cross-dependencies.

2.3 Standard exploratory analysis techniques

Simple exploratory analysis tools include graphical displays for examining the shape of the sample distribution such as bar charts and histograms, two-way tables such as contingency tables, and quantile plots (Amant and Cohen 1995). More complex exploratory analysis tools include discretization and scale conversions (Maimon and Rokach 2010, Liu et al. 2002), cluster analysis (Everitt et al. 2011), and correspondence analysis (Greenacre and Blasius 2006, Beh and Lombardo 2014). Here, we give a brief overview of these techniques.

2.3.1 Discretization

Discretization is a data processing procedure that transforms quantitative data into qualitative data and can also be useful if the variable is categorical but with too many categories or highly varying frequencies. This process is an important task of the data pre-processing, not only because some learning methods do not handle numerical variables, but also because discrete data and especially intervals are cognitively easier to apprehend and are often more relevant for a human interpretation than the actual values a variable takes (Liu et al. 2002). In addition to easier interpretation, the computation time can be significantly reduced when the data is transformed to a finite set of categories instead of containing a hypothetically infinite number of values as is the case for a numerical variable (Ho and Scott (1997), Frank and Witten (1999), Catlett (1991)). This computation time reduction is especially relevant if the cutting points are relevant to the learning problem at hand (Liu et al. (2002), Mittal and Cheong (2002)). Finally, in addition to harmonizing the nature of the data if it is heterogeneous, discretization can reveal complex relations between the variables to the learning process (Chen et al. (2017), Friedman and Goldszmidt (1996)). However, two key problems in association with discretization are how to select the number of intervals, and how to perform the discretization (see Maimon and Rokach (2010) and Liu et al. (2002) for more details on discretization methods).

Examples of discretization techniques include equal-width and equal-frequency discretization (Liu et al. 2002, Anderberg 1973), entropy and minimum description length based discretization (Friedman and Goldszmidt 1996, Fayyad and Irani 1993).

2.3.2 Clustering

Clustering is by far the most widely used exploratory analysis technique. The goal of clustering is to find a summary of the data in the form of groups of instances. That is to find an optimal grouping for which the instances within each cluster are *similar*, but the clusters are dissimilar to each other. The similarity between the instances is measured using all the measured attribute values. Hence, overall, the objects within the same cluster are assumed to behave similarly with respect to all the measured attributes.

However, in spite of its wide use, data clustering is a challenging task as it involves many choices (Jain et al. 1999). For example, aside from the data representation, the major choices in performing a cluster analysis include the choice of the variables (Anderberg 1973), the structure and type of the desired clustering (hierarchical, partitioning, density based, grid based, hard, fuzzy, ...). Furthermore, many techniques require choosing a similarity or dissimilarity measure between the items to cluster. Other clustering methods use a within- and between-cluster variability as a preliminary measure for clustering optimality (Rencher 2002). Moreover, depending on the type of clustering and the similarity measure, a variety of different methods can be used to perform a data clustering. Besides, very often, the number of clusters need to be specified and there is a multitude of cluster evaluation criteria (see Jain et al. (1999), and Maimon and Rokach (2010) for a comprehensive review of data clustering methods).

Because of the existence of several alternative ways to perform a clustering, and given the lack of consensus on a natural metric to evaluate a clustering, finding an appropriate data clustering is a complex and challenging task. Furthermore, another level of complexity arises when the data contains mixed-type variables since most clustering techniques are designed for uni-type data. To cluster mixed data, one of the most common approaches is converting the data set to a single data type, and applying standard clustering technique to the transformed data (Foss et al. 2016). However, this raises the same issues raised above for discretization (Section 2.3.1), namely how to select the number of intervals and how to perform the discretization.

Examples of clustering techniques include: K-means, self-organizing maps, spectral clustering, density based clustering (Jain et al. 1999).

2.3.3 Dimensionality reduction

A common problem encountered by most of the traditional clustering techniques is the *curse of dimensionality* (Maimon and Rokach 2010) which refers to the fact that increasing the number of attributes describing the objects

quickly leads to significant degradation of the performance of object clustering techniques (Murphy 2012). In fact, according to Maimon and Rokach (2010), it has been estimated that, as the number of dimensions (variables) increases, the number of instances (sample size) needs to increase exponentially in order to have an effective estimate of multivariate densities. To minimize the effect of high number of variables, dimensionality reduction techniques have been proposed.

Dimensionality reduction is a set of non-invertible mappings of data to a lower dimensional space (Maimon and Rokach 2010, Cunningham and Ghahramani 2015). The underlying hypothesis is that *although raw data is represented in a high dimensional format, the information contained in the data can be explained in a lower dimensional space.* The most commonly known dimensionality reduction techniques are linear, projection based, mappings where the goal is to find an optimal low-dimensional projection of the data (Sun et al. 2009). Namely, these techniques try to maximize the data variance captured by the low-dimensional projection, or equivalently to minimize the reconstruction error of the original data from projected data.

Examples of these commonly used techniques include principal component analysis (PCA) for numerical variables and multiple correspondence analysis (MCA) for categorical variables (Rencher (2002), Maimon and Rokach (2010)). Ultimately, the goal of these techniques is to uncover the associations between the objects and the features. That is to find the principal dimensions that capture the most variance possible, allowing for lower-dimensional description of the data (Saporta 2006). Other examples of linear mapping based dimensionality reduction techniques include Fisher's linear discriminant analysis LDA (Fisher 1936, Bishop 2006) where the purpose is to project the data such that the separation between classes is maximized. Non linear techniques include the non-negative matrix factorization NMF (Lee and Seung 2011), detailed in Section 2.5.2 (see la Torre (2012) and Scholkopf and Smola (2001) for examples of other nonlinear techniques).

Thanks to their capability of providing a description of the data in a lower dimensional space, these techniques have been shown to be particularly useful when the observed raw data is high dimensional data, but the intrinsic *information* included in the data can be visualized and explained in a lower dimensional subspace. Hence, they have been extensively used for visualizing high dimensional data. A common practice is thus to combine dimensionality reduction with a clustering technique where only the lower dimensional representation of the data is clustered.

However, despite their popularity, the usage of dimentsonality reduction methodologies for overcoming the obstacle of high dimensionality has several drawbacks (Maimon and Rokach 2010). First, the assumption that a large set of input features can be reduced to a small subset of relevant features is not always true. In some cases, all the features (or at least a significant majority) are of equal importance to the information contained in the data, and removing some features will cause a significant loss of important information. Second, in some cases, even after eliminating a set of irrelevant features, the

researcher is left with relatively large numbers of relevant features which means that a post-analysis method (such as clustering) is required to actually visualize and analyze the data. Furthermore, these methods have been shown to be noise sensitive and, although they provide interesting results when applied to relatively small data sets, they stay of limited use for the analysis of large data sets.

2.4 Co-clustering

Most of the data analysis literature focuses on the problem of clustering for structure extraction or a combination of a clustering technique with dimensionality reduction. However, the data can contain patterns that may be hard to capture using a traditional clustering approach. For example, consider the dyadic data of documents and words represented by a matrix, whose rows correspond to documents, columns correspond to words, and entries correspond to the counts of the words in documents. Given such data, one can perform a clustering of documents or words (depending on the goal of the analysis) using a traditional clustering approach. However, such one-sided clustering might fail to discover subtle patterns of the data. For example, some words may only appear in some sets of documents and inversely some documents may be clustered together because they contain specific words, which means that the data matrix exhibits a strong dependency structure between groups of words and groups of documents. In order to extract such patterns, one approach that has gained increasing attention, over the years, is the simultaneous clustering of the set of rows and the set of columns of the data matrix (also called co-clustering, cross-clustering or bi-clustering).

Proposed by Good (1965), then by Hartigan (1975), as an extension of standard clustering, co-clustering is a data mining technique that aims to jointly cluster both the object and feature dimensions simultaneously. Thus, taking advantage of the duality and interdependence between the set of objects and the set of variables. Whereas the principle of standard clustering is that of grouping objects that are similar with respect to the set of variables, the task of co-clustering is to *simultaneously* find groups of similar objects (with respect to the variables) and groups of similar variables (with respect to the objects).

The main advantage of these techniques is that they provide a powerful tool for extracting the existing dependencies between the instances and their descriptive variables, which enhances the interpretability of the clusters of instances using the clusters of variables and vice-versa. In some sense, this can be seen as a dimensionality reduction that operates both on the dimension of instances and the dimension of variables. Co-clustering is, for example, an interesting technique to consider in market analysis where a customer is represented by a vector, across a list of products (and vice-versa). In this case, the analyst can be more interested in identifying the subsets of customers that tend to buy the same subset of products and which products

they buy, than simply trying to group customers (or products) based on buying/selling patterns, which is the task accomplished by regular clustering. In contrast to a regular clustering technique, co-clustering customers and purchased products allows to discover the items of interest for a particular client/set of clients and thus build more precise recommendations and efficient promotions and sales strategies.

Another advantage of these techniques is their capability of summarizing a data matrix where the summary matrix is the matrix of co-clusters (also called blocks). The matrix of co-clusters is, in some sense, the essence of the data. This point of view is particularly interesting in exploratory data analysis where replacing the original, often large, data matrix by the considerably smaller matrix of co-clusters can facilitate analysis.

2.4.1 Definition

Let $\mathbf{X}$ be a data matrix as defined in Section 2.2 and let $\mathcal{U}$ be the set of I objects and $\mathcal{X}$ the set of K variables. Formally, most co-clustering methods are defined by a mapping $\hat{C}_{\mathcal{U}} : \mathcal{U} \to \{C_1^i, \ldots, C_g^i\}$ from the set of instances to groups of instances and a mapping $\hat{C}_{\mathcal{X}} : \mathcal{X} \to \{C_1^v, \ldots, C_m^v\}$ from the set of variables to groups of variables, where g and m are the number of clusters of instances and the number of clusters of variables, respectively. The intersection of a group of instances and a group of variables forms a co-cluster which can be seen as a *sub-matrix of the instance-variable matrix*.

The challenge of co-clustering is to extract a *structure* in the form of *homogeneous* blocks. The nature of such structure and the definition this *homogeneity* condition depend on the co-clustering method and can be characterized by how the rows and columns are assigned to clusters, and by the input data-type.

2.4.2 The homogeneity condition

The homogeneity condition, defined by the content of the co-cluster, varies from one method to another. Most co-clustering methods try to find co-clusters with constant values per co-cluster or, in the probabilistic context, co-clusters whose elements are issued from the same probability distribution. More generally, the literature distinguishes between co-clustering methods with respect to the content of the co-clusters (non exclusive examples are given in Table 2.1):

- Co-cluster with constant values (Table 2.1a),
- Co-cluster with constant values per row (Table 2.1b),
- Co-cluster with constant values per column (Table 2.1c),
- Co-cluster with coherent evolution over the rows (Table 2.1d),
- Co-cluster with coherent evolution over the columns (Table 2.1e),

- Co-cluster with coherent evolution over both rows and columns (Table 2.1f),
- Co-cluster with coherent values, obtained via a multiplicative or additive relationship between the row and column values or following a complex mathematical model that depends on the co-cluster. For example, in Table 2.1g, an element b_{kl} of the co-cluster is given by the additive model $b_{kl} = \mu + \alpha_k + \beta_l$ where $\mu = 5$, $\alpha = (4, 2, 3)$ and $\beta = (1, 5, 3)$.

1	1	1
1	1	1
1	1	1

(a)

1	1	1
3	3	3
2	2	2

(b)

1	3	2
1	3	2
1	3	2

(c)

1	3	9
5	20	80
3	15	75

(d)

1	6	3
5	24	9
25	96	18

(e)

1	3	9
3	15	75
5	20	80

(f)

10	19	12
8	12	10
9	13	11

(g)

Table 2.1 – *Examples of co-cluster types.*

The set of blocks form a structure. However, many different structures exist in the literature as each co-clustering method searches for a specific structure of blocks. In the following, we will focus on the methods that provide co-clusters with constant or coherent values to explore the most commonly extracted structures.

2.4.3 The co-clustering structure

The co-clustering structure is defined by the relationship between different clusters. More precisely, let us distinguish four types of methods.

1. Partitioning methods: the clusters define a partition of non empty non intersecting subsets that span the set of possibilities (the full set of I instances and the full set of K variables). In such cases, the co-clusters can be formed of the cartesian product of a partition of rows and a partition of columns (this is known as block clustering).
2. Overlapping clustering: each instance and each variable can belong to multiple clusters.
3. Nested clustering: the co-clusters can be defined by intersecting clusters. However, unlike overlapping clusters, if two clusters intersect, then one of them is necessarily a subset of the other (Mechelen et al. 2004). An example of nested clustering is given by the hierarchical co-clustering where the row and column clusters are defined by the cartesian product of a hierarchy of rows and a hierarchy of columns or a hierarchy of the $I \times K$ values in the data matrix.
4. Other types of structures include sets of subgroups. That is, a co-cluster is defined by a subgroup of instances and a subgroup of variables or in the case of value clustering, a subgroup of the $I \times K$ values.

However, in this case, not all rows/columns or values are required to belong to clusters.

Madeira and Oliveira (2004) provide a more detailed and general classification of these structure types found in gene expression co-clustering methods as shown in Figure 2.1. The details of these structures are as follows.

(a) Single co-cluster (Figure 2.1a), defined by one group of instances and one group of variables.

(b) Exclusive row and column co-clusters (Figure 2.1b): the assumption is that there exists a permutation of the rows and columns of the data matrix after which the co-clusters form rectangular diagonal blocks. This corresponds to a particular case of partition of the rows and a partition of the columns in Figure 2.1c.

(c) Non-overlapping co-clusters with checkerboard structure (Figure 2.1c): there exists a permutation of the matrix rows/columns after which the co-clusters form rectangular contiguous blocks. This corresponds to a partition of the rows and a partition of the columns.

(d) Exclusive-rows co-clusters (Figure 2.1d): this corresponds to co-clusters defined by a partition of the rows and overlapping clusters of columns.

(e) Exclusive-columns co-clusters (Figure 2.1e): this corresponds to co-clusters defined by overlapping clusters of rows and a partition of the columns.

(f) Non-Overlapping co-clusters with tree structure (Figure 2.1f).

(g) Non-Overlapping non-exclusive co-clusters (Figure 2.1g).

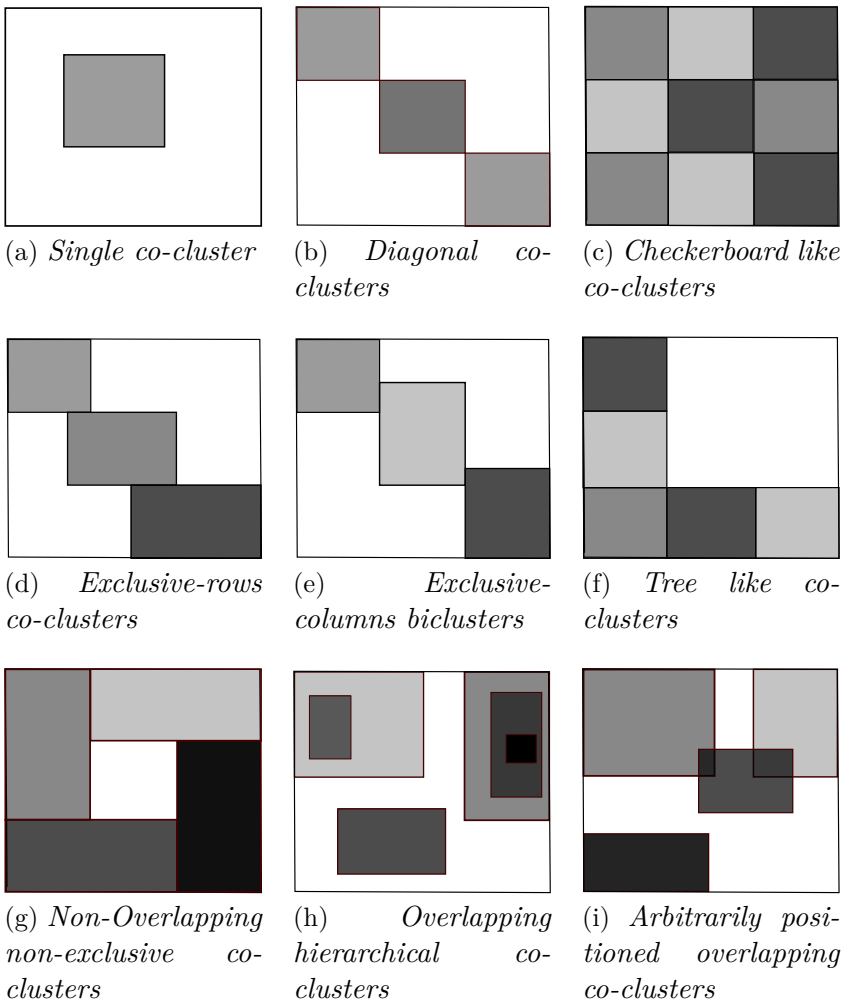
(h) Overlapping co-clusters with hierarchical structure (Figure 2.1h): hierarchical partitioning of the $I \times K$ values of the data matrix.

(i) Arbitrarily positioned overlapping co-clusters (Figure 2.1i): a set of possibly overlapping subgroups of the $I \times K$ values.

2.4.4 The co-clustering strategy

In order to extract the desired co-clustering structure, many co-clustering strategies have been studied in the literature. The most commonly followed strategies include the following.

- Clustering rows then columns independently, using a standard clustering technique, then simultaneously analyzing the results to fetch for a co-clustering structure (Lerman and Leredde 1977, Madeira and Oliveira 2004).



(a) *Single co-cluster* (b) *Diagonal co-clusters* (c) *Checkerboard like co-clusters*

(d) *Exclusive-rows co-clusters* (e) *Exclusive-columns biclusters* (f) *Tree like co-clusters*

(g) *Non-Overlapping non-exclusive co-clusters* (h) *Overlapping hierarchical co-clusters* (i) *Arbitrarily positioned overlapping co-clusters*

Figure 2.1 – *Examples of co-clustering structures.*

- Performing a standard clustering technique on the rows (resp. columns) then a standard clustering technique on the columns (resp. rows) while taking into account the first clustering results (Tishby et al. 1999). Examples of these methods include the Coupled Two-way Clustering (Getz et al. 2000), which performs a bi-clustering by alternating one-dimensional clustering algorithms, and the Interrelated Two-Way Clustering algorithm (Tang et al. 2001) that combines the results of one-way clustering(s) on both dimensions of the data matrix in order to produce co-clusters.

- Simultaneously clustering the rows and columns of data matrix (Govaert 1983; 1995, Kluger et al. 2003, Yoo and Choi 2010, Shan and Banerjee 2008).

In the coming sections, we introduce some of the most commonly known co-clustering methods to emphasize the richness of the field and to illustrate the different types of approaches. In particular, we will focus on the approaches that perform a simultaneous clustering of the rows and columns.

For a full survey of the co-clustering methods, readers are refered to: Charrad and Ahmed (2011), Tanay et al. (2005), Madeira and Oliveira (2004), Brault and Mariadassou (2015), Brault and Lomet (2015), Pontes et al. (2015), and Padilha and Campello (2017).

2.5 Simultaneous clustering of the instances and variables

The simultaneous clustering problem has been shown to be an NP-hard problem (Tanay et al. 2002). In particular, an exhaustive search of the space of solutions is infeasible, which requires most of the existing methods to base their search on heuristic optimization procedures. The use of a suitable co-cluster evaluation measure and the development of an effective search heuristic are two crucial factors for finding significant co-clusters with reasonable resources. Pontes et al. (2015) reviews a large number of biclustering approaches used in gene expression analysis and classifies them into two categories: biclustering algorithm based on evaluation measures, and non metric-based biclustering algorithms.

In this section, we introduce some of the most common approaches to perform a simultaneous clustering of the instances and variables in a data table. In particular, we will distinguish between cost function, linear algebra, and parameter identification based methods.

2.5.1 Deterministic cost function based co-clustering

In the literature, many co-clustering algorithms propose to optimize an objective function called co-cluster evaluation function. Most of theses objective functions try to summarize the original data matrix $\mathbf{X}$ by a new smaller matrix $\hat{\mathbf{X}} = (\hat{x}_{kl})_{1\leq k\leq g, 1\leq l\leq m}$ containing a summarized representation of the blocks (such as the mean or median value), or by some reconstructed data matrix $\hat{\mathbf{X}} = (\hat{x}_{ij})_{1\leq i\leq I, 1\leq j\leq K}$ of the same size as $\mathbf{X}$ but with constant entries within each block.

Deterministic objective function based algorithms try to define the couple of mappings from the instances to instance clusters and from the variables to variable clusters that optimize a co-cluster quality measure that characterizes the difference between the original data $\mathbf{X}$ and the co-clustered data $\hat{\mathbf{X}}$.

Hartigan's direct clustering

When Hartigan (1972) introduced co-clustering as "direct clustering", he proposed to simultaneously cluster the rows and columns of a data table. The algorithm seeks co-clusters with constant values or low within-block variance. To do this, he approximates the original data matrix by the matrix $\hat{\mathbf{X}}$ that minimizes the sum of squared residues. As a result, the values within each co-cluster are identical. The quality of a co-cluster $B_{kl} = (C_k^i, C_l^v)$

(defined by the cluster C_k^i of rows and the cluster C_l^v of columns) is given by the within-co-cluster variance:

$$\mathcal{C}(C_k^i, C_l^v) = \sum_{i \in C_k^i, j \in C_l^v} (x_{ij} - \hat{x}_{ij})^2.$$

Given a desired number of bi-clusters B, the proposed algorithm is a divide and conquer type algorithm that starts with the entire data in a single block then at each iteration finds the row split or column split that produces the largest reduction of the total within-block variance. The splitting continues until the reduction of block variance is not greater than a given threshold. The algorithm results in a tree like hierarchical clustering of rows and columns of the data matrix. The quality of a co-clustering is measured by the overall variance of the B bi-clusters:

$$\mathcal{C}(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{b=1}^{B} \sum_{i=1}^{I} \sum_{j=1}^{K} (x_{ij} - \hat{x}_{ij})^2.$$

However, the main drawback of this successive splitting heuristic is that partitions cannot be reconsidered once they have been split. Hence, the final hierarchical clustering can miss some quality biclusters due to premature division of the data matrix. Also, the number of desired co-clusters need to be specified.

One block at a time using the Cheng and Church Algorithm

Cheng and Church (2000) were the first to propose a co-clustering algorithm for gene expression data. The approach uses a measure called Mean Squared Residue (MSR) as measure of co-cluster quality and a greedy algorithm for co-cluster extraction. In particular, the residue of an element x_{ij} to block $B_{kl} = (C_k^i, C_l^v)$ is defined as:

$$x_{ij} - x_{i,C_l^v} - x_{C_k^i,j} + x_{C_k^i,C_l^v},$$

where $x_{C_k^i,C_l^v} = \frac{1}{|C_k^i||C_l^v|} \sum_{i \in C_k^i, j \in C_l^v} x_{ij}$ is the average of all entries in the block B_{kl}, $x_{i,C_l^v} = \frac{1}{|C_l^v|} \sum_{j \in C_l^v} x_{ij}$ is the mean of all entries in row i whose columns belong to C_l^v, $x_{C_k^i,j} = \frac{1}{|C_k^i|} \sum_{i \in C_k^i} x_{ij}$ is the mean of all entries in column j whose rows belong to cluster C_k^i, and $\{|C_k^i|, |C_l^v|\}$ are the cardinalities of C_k^i and C_l^v respectively.

For a block B_{kl}, the goal is to find a sub matrix defined by the couple of groups (C_k^i, C_l^v) that minimizes, up to a certain threshold, the mean squared residue defined as:

$$\mathcal{C}(C_k^i, C_l^v) = \frac{1}{|C_k^i||C_l^v|} \sum_{i \in C_k^i, j \in C_l^v} (x_{ij} - x_{i,C_l^v} - x_{C_k^i,j} + x_{C_k^i,C_l^v})^2.$$

This mean squared residue measures the level of coherence within the co-cluster as the difference between the observed values x_{ij} and the expected values predicted from the corresponding row mean, column mean and bi-cluster mean (Madeira and Oliveira 2004).

The approach uses a greedy iterative search algorithm for rows and columns suppression while minimizing the objective function $\mathcal{C}$ up to a given threshold. The algorithm produces one co-cluster at a time (as in Figure 2.1a) and is composed of two stages. In the node (row or column) deletion stage, the algorithm starts with one co-cluster containing the original data matrix then proceeds in iteratively removing rows and columns to achieve the largest decrease of the score while keeping its value under a threshold value. This requires the computation of the scores of all the sub-matrices that may be the consequences of any row or column removal, before each choice of a row/column removal can be made. Once the threshold is reached, the second stage consists of adding rows and columns back to the block if this can be done without increasing the score. After each co-cluster is produced, the elements corresponding to the co-cluster are replaced with random numbers, then the same procedure is applied on the modified matrix to generate another, possibly overlapping, co-cluster until the required number of co-clusters is reached. Within a co-cluster, the low mean squared residue condition enables extracting co-clusters with coherent values and also constant values in some cases. The final extracted structure would reassemble to that in Figure 2.1i.

The algorithm presents several drawbacks, the most important of which is the use of a threshold parameter for rejecting solutions, which is dependent on each data set. Also, the algorithm produces only one co-cluster at a time and as the algorithm proceeds, the random numbers used as replacements for the co-clusters can interfere with the future discovery of co-clusters, especially ones that overlap with the discovered ones which is addressed by Yang et al. (2003). Yang et al. (2003) propose an algorithm called FLexible Overlapped biClustering (FLOC) that simultaneously produces B co-clusters whose mean residues are all less than a predefined constant, without the impact of random interference.

Extracting B-blocks simulatenously

Cho et al. (2004) also use squared residue measures similar to those of Hartigan (1972) and Cheng and Church (2000) in k-means like co-clustering called Minimum Sum-Squared Residue Coclustering (MSSRCC) for homogeneous block extraction. An homogeneous block is defined by a sub matrix having low average square residues. For every element x_{ij} that may belong to co-cluster B_{kl}, they define two measures for co-cluster quality:

$$H_{B_{kl}} = (h_{ij})_{1\leq i\leq I, 1\leq j\leq K} \text{ where } h_{ij} = x_{ij} - x_{C_k^i, C_l^v},$$

and

$$H_{B_{kl}} = (h_{ij})_{1\leq i\leq I, 1\leq j\leq K} \text{ where } h_{ij} = x_{ij} - x_{i,C_l^v} - x_{C_k^i, j} + x_{C_k^i, C_l^v},$$

where $x_{C_k^i,C_l^v}$, x_{i,C_l^v} and $x_{C_k^i,j}$ are as defined above. For both measures, they propose an algorithm to minimize the total squared residues:

$$||H||_2^2 = \sum_{k=1}^{g}\sum_{l=1}^{m}\sum_{i\in C_k^i, j\in C_l^v} h_{ij}^2.$$

The first score measures the sum of squared differences between each entry in the co-cluster and the mean of the co-cluster, producing co-clusters with low variance or constant values (as in Hartigan (1972)). The second score measures the sum of squared differences between each entry in the co-cluster and the corresponding row mean and the column mean, while counting for the co-cluster mean for symmetry (as in Cheng and Church (2000)). For co-cluster extraction, the authors propose two iterative algorithms that monotonically decrease the objective functions and converge to a local minimum.

The main difference with Cheng and Church (2000) is that Cho et al. (2004) extract B co-clusters simultaneously while Cheng and Church (2000) extracts one co-cluster at a time. Cho and Dhillon (2008) provide specific strategies to enhance the performance of the MSSRCC. For example, like ordinary k-means-type clustering algorithms, the approach suffers from being trapped in local minima and generating empty clusters. Cho and Dhillon (2008) try to resolve these problems by adopting an incremental local search (LS) strategy, where incremental moves of rows and columns among clusters are performed in order to decrease the objective function value. Also, different data pre-processing transformations and cluster initialization strategies are investigated. Anagnostopoulos et al. (2008) propose a generalization of this approach that minimizes a p-norm of the residue matrix $H = (h_{ij})$.

The CRO methods

In the same context of cost function opitimization, Govaert (1983) proposes three algorithms: Crobin, Croeuc and Croki2 for binary, continuous and contingency data. Denote $\mathbf{z}_{(I\times g)}$, $\mathbf{w}_{(K\times m)}$ and $\hat{\mathbf{X}}_{(g\times m)}$ the partition of rows, the partition of columns and the summary matrix. The algorithms alternate between finding the row partition and finding the column partition until the co-clustering criterion reaches a local optimum.

For binary data, the Crobin algorithm searches for homogeneous blocks (blocks with majority of ones or majority of zeros) by optimizing the criterion:

$$\mathcal{C}(\mathbf{z},\mathbf{w},\hat{\mathbf{X}}) = ||\mathbf{X} - \mathbf{z}\hat{\mathbf{X}}\mathbf{w}^t||_1 = \sum_{k=1}^{g}\sum_{l=1}^{m}\sum_{i\in C_k^i, j\in C_l^v} |x_{ij} - \hat{x}_{kl}|,$$

where $\hat{\mathbf{X}} = (\hat{x}_{kl})_{1\le k\le g, 1\le l\le m}$ and $\hat{x}_{kl} \in \{0,1\}$.

For continuous data, the Croeuc algorithm uses alternated k-means algorithm to minimizes the squared euclidean distances between the elements

in the block B_{kl} and its characterizing value $\hat{x}_{kl}$:

$$\mathcal{C}(\mathbf{z}, \mathbf{w}, \hat{\mathbf{X}}) = ||\mathbf{X} - \mathbf{z}\hat{\mathbf{X}}\mathbf{w}^t||^2 = \sum_{k=1}^{g}\sum_{l=1}^{m} \sum_{i \in C_k^i, j \in C_l^v} (x_{ij} - \hat{x}_{kl})^2.$$

It is worth noting that these algorithms optimize a criterion that is data-type dependent and that their best bet is to achieve a local optimum of the objective function. Furthermore, the number of clusters of rows and the number of clusters of columns need to be specified. The Croki2 algorithm will be addressed in Section 2.6.1.

2.5.2 Linear algebra and matrix reconstruction based co-clustering

While the methods introduced above try to optimize the difference between the original data matrix and a summary data matrix, other co-clustering methods focus on decomposing the original matrix to extract associated clusters. Among these techniques, we mention those based on latent matrices like Non-negative Matrix Factorization (NMF), and those based on applying a dimensionaliy reduction procedure followed by a standard clustering technique.

Matrix reconstruction based co-clustering

Matrix reconstruction based methods try to re-write the optimization (co-clustering) problem in the form of matrix approximation problem, and use matrix factorization (Lee and Seung 2011, Yoo and Choi 2010) to fetch for co-clusters. These algorithms include non-negative matrix factorization (NMF), non-negative tri-factorization (NTF) and non-negative block value decomposition (NBVD).

Non-negative matrix factorization (NMF) searches for the decomposition of a *non negative matrix* $\mathbf{X}$ into a product of two *non negative latent matrices* that are used for row and column clustering. For example, Lee and Seung (2011) optimize a cost function that characterizes the difference between the original matrix $\mathbf{X}_{(I\times K)}$ and the product of two matrices $\mathbf{z}_{(I\times g)}$, and $\mathbf{w}^T_{(K\times g)}$ (^{T} is the transpose). The proposed cost function is either the least square euclidean distance or a generalized divergence measure D of $\mathbf{X}$ from $\mathbf{z}\mathbf{w}^T$

$$\underset{\mathbf{z}\geq 0, \mathbf{w}\geq 0}{\operatorname{argmin}} \; ||\mathbf{X} - \mathbf{z}\mathbf{w}^T||^2,$$

or

$$\underset{\mathbf{z}\geq 0, \mathbf{w}\geq 0}{\operatorname{argmin}} \left(D(\mathbf{X}||\mathbf{c} = \mathbf{z}\mathbf{w}^T) = \sum_{i,j} (x_{ij} \log \frac{x_{ij}}{c_{ij}} - x_{ij} + c_{ij}) \right),$$

where $||.||$ is the Frobenius matrix norm. Iterative minimization algorithms are used to find local minima and are based on multiplicative updating rules.

Given the latent matrices $\mathbf{z}$ and $\mathbf{w}$, the clustering of rows can be performed by considering the columns of $\mathbf{z}$ as row cluster centroids and the i^{th} row of the data matrix $\mathbf{X}$ can be associated to the centroid c_i (i.e., to its corresponding cluster) which gives the maximum contribution in the linear combination

$$c_i = \underset{k}{\operatorname{argmax}}\, \mathbf{w}_{ki},$$

and inversely for obtaining clusters of columns (Buono and Pio 2015).

Buono and Pio (2015) note that this basic NMF provides only *casual clustering* and that, to obtain a solution that guarantees a real clustering interpretation, additional orthogonality constraints on $\mathbf{z}$ and/or $\mathbf{w}$ should be imposed as in Ding et al. (2006). In Ding et al. (2006), the Frobenius based optimization problem is modified to generate a clustering of rows, by imposing the orthogonality constraint on $\mathbf{w}$ ($\mathbf{w}\mathbf{w}^T = \mathbf{1}$, the identity matrix). In the same manner, a clustering of columns can be achieved by imposing an orthogonality constraint on $\mathbf{z}$ ($\mathbf{z}^T\mathbf{z} = \mathbf{1}$).

To perform simultaneous clustering of rows and columns, orthogonality constraints over $\mathbf{z}$ and $\mathbf{w}$ need to be met simultaneously (solving the optimization problem under the constraints $\mathbf{w}\mathbf{w}^T = \mathbf{1}$ and $\mathbf{z}^T\mathbf{z} = \mathbf{1}$). This condition is too restrictive, according to Buono and Pio (2015). Furthermore, under this double orthogonality constraint, the solutions result in a rather poor low-rank approximation of the data matrix $\mathbf{X}$ (always according to Buono and Pio (2015)). Hence, for better approximation, a third latent matrix can be introduced, to create non-negative tri-factorization (NTF), which allows the low-rank approximation to remain accurate, while a soft-orthogonality of $\mathbf{z}$ and $\mathbf{w}$ is maintained (Buono and Pio 2015).

Non-negative tri-factorization algorithms are similar to NMF except they use only the Frobenius norm and they search to decompose $\mathbf{X}$ into the product $\mathbf{z}\hat{\mathbf{X}}\mathbf{w}^T$ of three latent matrices where $\hat{\mathbf{X}}$ is the non negative summary matrix and the two binary matrices $\mathbf{z}$ and $\mathbf{w}$ are for rows and columns classification respectively (Yoo and Choi 2010). Just like NMF, the tri-factorization algorithm converges to a local minimum.

Non-negative block value decomposition (NBVD) has the same goal as the NTF which is factorizing the data matrix into three latent matrices but uses fuzzy classification matrices for the rows and columns (Long et al. 2005). The algorithm converges to a local minimum by iteratively updating the decomposition matrices using a set of multiplicative updating rules.

Note that these methods apply only to numerical non negative matrices. Also, the NMF approach requires the number of row and column clusters to be the same while NTF allows for the numbers of clusters to differ. However, in both cases, the number of clusters need to be specified. Note also that the Croeuc, Crobin and Croki2 algorithms we mentioned earlier for continuous, binary and contingency data (Govaert 1983) can be reformulated as matrix decomposition algorithms.

Spectral co-clustering

Spectral clustering is a technique commonly used in graph theory to extract clusters in edge-weighted graphs using the spectrum of a Laplacian matrix. The idea is to reduce the dimensionality of the original data in a manner that best separates clusters (if they exist), then one can apply a standard clustering technique in the new lower dimension to extract them. A technique of choice is singular value decomposition of the graph Laplacian. The reduction in dimensionnality is performed by computing the first, say l, eigenvectors of the Laplacian which will constitute the reduced matrix on which a standard clustering technique can be performed. This technique has been extended to perform clustering of non graph-data by building a graph in which nodes correspond to data points and links are related to the similarity between the data points.

Dhillon (2001) proposes a spectral co-clustering algorithm to co-cluster a document-word matrix by modeling it by a bipartite graph. The document collection is first represented by a word-by-document matrix of weights A whose rows correspond to words and columns to documents and the entries of which correspond to the number of appearances of the i^{th} word in the j^{th} document. The word-clusters and document-clusters can then be found by performing a singular value decomposition of the normalized matrix A_n given by:

$$A_n = D_1^{-1/2} A D_2^{-1/2}, \tag{2.1}$$

where D_1 and D_2 are the weights of documents and words expressed in diagonal matrices such that: $D_{1_{ii}} = \sum_j A_{ij}$ and $D_{2_{jj}} = \sum_i A_{ij}$.

The right singular vector gives a bi-partitioning (two clusters) of documents while the left singular vector gives a bi-partitioning of words. The algorithm perform a bi-partitionning of the documents and words by applying a k-means algorithm on the reduced matrix given by:

$$z = \begin{pmatrix} D_1^{-\frac{1}{2}} u \\ D_2^{-\frac{1}{2}} v \end{pmatrix},$$

where u and v are the second left and right singular vectors u and v, respectively.

To obtain B clusters of documents and B clusters of words (called multi-partitioning), the authors state that one possibility is to recursively apply the bi-partitioning algorithm until reaching the number of clusters desired or construct a reduced data matrix using more singular vectors then apply a k-means like algorithm. To know how many vectors to choose, the authors suggest using $l = \lceil \log_2(B) \rceil$ right and left singular vectors. Let U be the matrix containing the l left singular vectors and V the matrix containing the l right ones, create the l-dimensional reduced data

$$Z = \begin{pmatrix} D_1^{-\frac{1}{2}} U \\ D_2^{-\frac{1}{2}} V \end{pmatrix},$$

and run a k-means algorithm (on Z). The l-dimensional reduced data often contains k-modal information about the data set. However, this provides no guarantees on how many singular vectors to use when seeking B clusters. Furthermore, the algorithm only extracts partitioning structures with a number of row-clusters equal to the number of column-clusters, and with the restriction that each document cluster is associated to a word cluster.

Dhillon (2001) illustrated the problem with term-document matrices but the same can be applied to market baskets or genes expression data. For example, in their work, Kluger et al. (2003) extend the work of Dhillon (2001) and propose an algorithm that performs co-clustering of gene expression data while allowing the number of row clusters to be different from that of column clusters.

Possibilistic Spectral Biclustering

Cano et al. (2007) proposed the Possibilistic Spectral Biclustering algorithm (PSB) which is based on the use of Singular Value Decomposition together with two one-dimensional clusterings. However, unlike the approach proposed by Dhillon (2001), each combination of the resulting clusters of rows and clusters of columns is a candidate co-cluster. Each candidate co-cluster will be post-processed in order to improve its quality when possible, or rejected if it is not considered a quality solution (Pontes et al. 2015).

Yang et al. (2011) have also proposed a strategy similar to the one of Cano et al. (2007). The approach uses SVD to decompose the input expression matrix into a group of centroid rows (genes) and a group of centroid columns (conditions). After centering the rows of these two basis matrices, clustering is applied to both of them by a mixed clustering algorithm, based on agglomerative hierarchical clustering and on the use of the sub-matrix correlation score as dissimilarity measure. Here too, every pair of the resulting row and column clusters form a candidate co-cluster. Like in Cano et al. (2007), a final post-processing step is executed in order to obtain inclusion-maximal co-clusters, through merging of the clusters. The authors state that, unlike the standard spectral biclustering approach proposed by Kluger et al. (2003), they utilize the eigenvectors corresponding to the eigenvalues that account for most of the energy as opposed to using only the eigenvectors corresponding to the first two or three eigenvalues. The approach generates possibly overlapping co-clusters.

However, the application of the SVD to large and high-dimensional data is unfeasible since it requires a computational time that is quadratic in the data size ($\mathcal{O}(min(IK^2, I^2K))$ for a matrix $I \times K$). Hence, the applicability of both possibilistic and standard spectral co-clustering to large and high dimensional data sets is very limited due to the required SVD. Furthermore, the SVD input matrix must be complete with no missing values.

2.5.3 Probabilistic model based co-clustering

Inspired by the use of finite mixture models for standard clustering, probabilistic co-clustering methods (e.g., Dhillon et al. (2003), Banerjee et al. (2007), Govaert and Nadif (2008), Wang et al. (2009), Shan and Banerjee (2010)) are based on the assumption that data was generated as a mixture of probability density functions and their goal is to estimate the parameters of the underlying distributions and the posterior probabilities of each co-cluster given the data.

Latent Block Model

Latent Block Model is a co-clustering technique that supposes the existence of a row partition and a column partition that can explain co-clusters in a checkerboard like structure (Govaert and Nadif 2008) (Figures 2.1c and 2.1b). It provides an extension of the stochastic block model (SBM), which is a mixture model often used for clustering the nodes in networks (Snijders and Nowicki 1997). The idea is to use mixture models to discover the latent structure as homogeneous blocks. In this context, a co-cluster is said to be homogeneous when all its elements are realizations of a probability distribution that depends only on the block itself.

A mixture model based clustering is obtained by assuming that the data is generated from a mixture of densities:

$$f(\mathbf{X};\theta) = \prod_{i} \sum_{k=1}^{g} \pi_k \, \varphi_k(x_{i.};\alpha_k), \tag{2.2}$$

with: $\pi_k \in]0,1[\; \forall \, k$ and $\sum_{k=1}^{g} \pi_k = 1$, π_k the probability that the row $x_{i.}$ belongs to the k^{th} cluster, g the number of components in the mixture, φ_k the probability density of the k^{th} component, α_k the set of all parameters of the k^{th} component, and θ is a vector containing all parameters of the model: $\theta = (\pi_1, \ldots, \pi_g, \alpha_1, \ldots, \alpha_g)$. The clustering of rows into g components can be found by estimating the parameters of the model.

For the co-clustering context, let $\mathbf{z}$ be the binary row partition matrix and $\mathbf{w}$ be the binary column partition matrix with $\mathbf{z}_{ik} = 1 \iff x_{i.} \in C_k^i$ and $\mathbf{w}_{jl} = 1 \iff x_{.j} \in C_l^v$. The density function of the observed data $\mathbf{X}$ can be given by:

$$f(\mathbf{X};\theta) = \sum_{(\mathbf{z},\mathbf{w}) \in (\mathcal{Z} \times \mathcal{W})} p((\mathbf{z},\mathbf{w});\theta) f(\mathbf{X}|\mathbf{z},\mathbf{w};\theta)), \tag{2.3}$$

where $(\mathcal{Z} \times \mathcal{W})$ is the set of all possible partitions, and θ the set of all model parameters.

The latent block model (Govaert and Nadif 2003) is based on two main assumptions:

1. the row partition and column partition are independent,

2. given the row and column partitions, all elements in a co-cluster are independent realizations of a probability distribution that depends only on the latent variables.

Under these assumptions, the LBM's probability density can be written as:

$$f(\mathbf{X};\theta) = \sum_{(\mathbf{z},\mathbf{w})\in(\mathcal{Z}\times\mathcal{W})} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,l} \rho_l^{w_{jl}} \prod_{i,j,k,l} \varphi_{kl}(x_{ij};\alpha_{kl})^{z_{ik}w_{jl}},$$

where φ_{kl} is the block conditional probability distribution. The log-likelihood of this model is therefore given by

$$L(\theta) = \log(f(\mathbf{X};\theta)) = \log\Big(\sum_{(\mathbf{z},\mathbf{w})\in(\mathcal{Z}\times\mathcal{W})} \prod_{i,k} \pi_k^{z_{i,k}} \prod_{j,l} \rho_l^{w_{jl}} \prod_{i,j,k,l} \varphi_{kl}(x_{ij};\alpha_{kl})^{z_{ik}w_{jl}}\Big).$$

For parameter estimation, Govaert and Nadif (2003) proposed using maximum likelihood approximation. Since then, maximum likelihood approximation algorithms for latent block models have been extensively used. These algorithms are based on alternating applications of an EM derivative algorithm (McLachlan and Krishnan 2008).

Naturally, Latent Block Models assume a conditional probability φ_{kl} that is data-type dependent. For example, φ_{kl} is considered Gaussian for continuous data (Govaert and Nadif 2009), Bernoulli for binary data (Govaert and Nadif 2003; 2008), Poisson for contingency data (Govaert and Nadif 2007) and Multinomial for discrete data (Keribin et al. 2013, Brault 2014). Therefore, their application to mixed data can be tricky.

In Appendix B, we propose an extension of this model which applies to data containing numerical and binary variables. However, while this extension provides a co-clustering of the mixed data, one can see, from the experiments provided in Appendix B, that its contribution stays limited to some extent, as it inherits the drawbacks of LBM. Namely, the number of clusters need to be known in advance. Since this is not always the case, choosing the right numbers of clusters is crucial for successful modeling both for the standard LBM and for the extended version. To tackle this problem, model selection criteria like the Bayesian Information Criterion BIC (Schwarz 1978), the Akaike Information Criterion AIC (Akaike 1974), and the Integrated Classification Criterion ICL (Biernacki et al. 2000), can be used to infer the optimal number of clusters in the standard LBM. However, these criteria have not been extended to the mixed data version.

Bayesian estimation of LBM

Shan and Banerjee (2008) proposed a Bayesian co-clustering model (BCC). The generative model is given by Figure 2.2. In fact, Bayesian co-clustering (Shan and Banerjee 2008) assumes that the model parameters are random variables with a prior distribution. The associated generative model is as follows: two separate Dirichlet distributions $Dir(\alpha_1)$ and $Dir(\alpha_2)$ from which

the probabilities of each row cluster C_k^i and column cluster C_l^v given each row $x_{i.}$ and column $x_{.j}$ are generated (denote these probabilities as z_1 and z_2). Row clusters for entries in row $x_{i.}$ and column clusters for entries in column $x_{.j}$ are then generated from a discrete distribution (usually Multinomial) with parameters π_1 and π_2 respectively (Shan and Banerjee 2008). Finally, each entry of the data matrix is generated according to the corresponding co-cluster which is assumed to have an exponential family distribution.

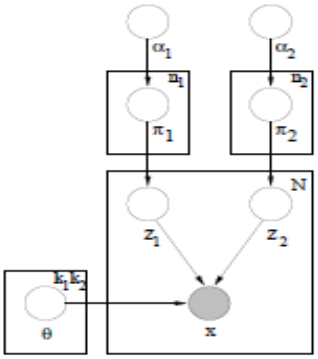


Figure 2.2 – *Bayesian co-clustering: data generative model.*

Given this generative model, the probability of observing the data matrix $\mathbf{X}$ can be written:

$$p(\mathbf{X}|\alpha_1, \alpha_2, \theta) =$$
$$\int_{\pi_{1_1}} \cdots \int_{\pi_{1_I}} \int_{\pi_{2_1}} \cdots \int_{\pi_{2_K}} \left(\prod_i p(\pi_{1_i}|\alpha_1) \right) \left(\prod_j p(\pi_{2_j}|\alpha_2) \right.$$
$$\left. \left(\prod_{i,j} \sum_{z_{1_{ij}}, z_{2_{ij}}} (p(z_{1_{ij}}|\pi_{1_i}) p(z_{2_{ij}}|\pi_{2_j}) p(x_{ij}|\theta_{z_{1_{ij}}, z_{2_{ij}}}))^{\delta_{ij}} d\pi_{1_1} \ldots \pi_{1_I} d\pi_{2_1} \ldots \ \pi_{2_K} \right) .$$

Assuming that the latent variables are independent and that the assignments of $z_{1_{ij}}$ and $z_{2_{ij}}$ for an entry x_{ij} are independent from the assignments of other entries, the authors use a variational EM like Bayesian algorithm to perform inference over an approximation of the latent variable distributions. The algorithm uses an approximation q of the true latent variable distribution $p(z_1, z_2, \pi_1, \pi_2|\alpha_1, \alpha_2, \Theta)$ denoted $q(z_1, z_2, \pi_1, \pi_2|\gamma_1, \gamma_2, \phi_1, \phi_2)$ where $\{\gamma_1, \gamma_2\}$ are called variational Dirichlet distribution parameters and $\{\phi_1, \phi_2\}$ are variational discrete distribution parameters. The E-steps estimates the values of γ_1, γ_2, ϕ_1 and ϕ_2 that maximize the log-likelihood with regards to parameters α_1, α_2 and θ while the M-step estimates α_1, α_2 and θ based on the previously estimated variational parameters.

In Wang et al. (2009), the authors extend the BCC model and propose using collapsed Gibbs sampling and collapsed variational inference instead, which enhances the accuracy the likelihood estimates.

Nonparametric bayesian co-clustering

Meeds and Roweis (2007) propose a probabilistic non parametric Bayesian co-clustering model where each cluster is part of a mixture having a non parametric fully Bayesian prior, namely the Pitman-Yor Process prior (Pitman 2002), which is a generalization of Dirichlet Processes that favors uniform cluster sizes and allows to count for the possibly infinite number of clusters. Thus, the model does not require the number of row and column clusters to be specified a priori. The algorithm returns a distribution over row and column partitions which are averaged according to their posterior probabilities. Markov Chain Monte Carlo (MCMC) is used for sampling. For missing value imputation, the predictions are averaged. For cluster analysis, the partitions are averaged by performing a symmetric neighborhood graph in which the weight of an edge between two nodes is the fraction of partitions in which they were found in the same cluster.

The non parametric BCC model differs from the basic BCC model in that, in the former, each row or column object is assigned to a single row or column cluster (resulting in a partitioning structure as in Figure 2.1c), whereas the latter samples distinct row and column cluster latent variables for each entry of the matrix, resulting in a possibly overlapping clustering of the data entries as in Figure 2.1i.

The infinite hidden relational model

The infinite hidden relational model IRM (Xu et al. (2006), Wang et al. (2012)) is an extension of the non-parametric Bayesian model that leverages features associated to the rows and columns of the contingency matrix to forecast relationships among previously unseen data. Ishiguro et al. (2012) proposes an extension of the IRM that applies to binary data and counts for the noise in the data and only a subset of rows and columns are used for co-clustering.

Kemp et al. (2006) propose a similar nonparametric bayesian approach for relational learning which can be seen as a co-clustering technique when the studied entities are the objects and features of a data table, and an entry represents the relation between the corresponding object and features. The model applies to binary data and extracts clusters of entities (clusters of objects and clusters of variables) simultaneously and qualifies the relationships between the clusters as manifested by the data. To do this, two independent Dirichlet processes are used as priors for the cluster partitions. Therefore, the model does not require the number of clusters to be specified in advance. This Infinite Relational Model is very similar to the non parametric Bayesian approach of Meeds and Roweis (2007), except that it can model not only binary relations between two different kinds of objects, but also binary relations between the same kind of objects. One drawback of IRM is the Chinese Restaurant Process (CRP) prior which allows for many very small clusters which raises the question whether these clusters are in fact clusters or simply a result of the underlying process.

CrossCat for high dimensional heterogeneous data

The CrossCat model (Mansinghka et al. 2016) is a fully Bayesian non-parametric method for analyzing heterogeneous, high dimensional data, through clustering the instances and variables. The model provides a fully Bayesian non parametric approach for density estimation, using Dirichlet processes as priors. In particular, CrossCat is based on approximate Bayesian inference in a hierarchical non parametric model for data tables, which consists of a hierarchical structure in which an outer clustering groups the variables in a set of non-overlapping views of independent variables. Then, each view is clustered independently at the level of instances using a separate Dirichlet process mixture. These inner mixture components are simple parametric models whose form depends on the types of data in the table (Mansinghka et al. 2016). Thus, the final structure extracted by CrossCat contains one clustering of the variables and multiple clustering(s) of the instances. This makes CrossCat rather flexible but also difficult to interpret.

In terms of flexibility, CrossCat provides a joint density estimation that simultaneously supports heterogeneous data types, detects independencies between the variables, and produces representations that support efficient prediction (Mansinghka et al. 2016). Hence, the model enables clustering multiple variables of different types in the same view, which lacks from most existing co-clustering methods. It detects independencies between the variables through the outer clustering which partitions variables into groups that are independent of one another. However, because this outer clustering operates at the variable level, it does not allow to identify complex and partial cross-dependencies between variables. Similarly to the non parametric model in Meeds and Roweis (2007), CrossCat uses averaged predictions for prediction and missing value imputation and constructs a similarity function in which the similarity between two rows and columns is represented by the fraction of partitions in which they were found in the same cluster.

In terms of interpretability, the model provides approximated posterior samples. Each sample provides an estimate of the full joint distribution. This feature contributes to its flexibility and its efficacy for prediction and missing value imputation, because prediction is based on averaged calculation or sampling from the conditional densities implied by each sample. However, in terms of interpretability of the results, analyzing the full space of possibilities or multiple independent conclusions about the data rather complicates the task of exploratory analysis.

In terms of scalability, CrossCat has a complexity in $\mathcal{O}(IK\tau\sigma)$ where I is the number of instances, K is the number of variables, τ is the maximum number of variable clusters (a.k.a. views) and σ maximum number of instance clusters. In practice, $\tau = \mathcal{O}(\sqrt{K})$ and $\sigma = \mathcal{O}(\sqrt{I})$ are reasonable assumptions. Thus the scalability of CrossCat enables handling data with millions of entries. However, beyond its computational complexity, the effective computational time of CrossCat is quite large, since its main transition in the MCMC algorithm runs from 10^3 to 10^5 times.

Summary. In summary, both metric and probabilistic approaches are known to have their advantages and limitations: despite being quite efficient in modeling data issued from virtually any distribution, probabilistic methods are computationally demanding and hardly scalable. Metric methods on the other hand are less computationally demanding but present the need to choose the "right" distance that uncovers the underlying latent co-clusters structure based on available data (Laclau et al. 2017).

2.6 Simultaneous clustering of two categorical variables

In the previous section, we introduced co-clustering approaches that apply to data where a set of objects is described by a set of variables. In other words, the rows and columns of the data table refer to different set of entities, with different roles. In this section, we introduce a special type of co-clustering approaches in which an object is described by its relationship with other objects. These other objects can be of the same nature as the first set, as in the case of graphs or of a different nature, as is the case of contingency tables and bipartite graphs. These types of data are often referred to as relational or contingency data. Co-clustering such data consists of a simultaneous clustering of the values of two variables instead of a simultaneous clustering of a set of instances and a set of variables. In the following, we focus on co-clustering contingency data formed by two categorical variables to set the ground for the methodology detailed in the next chapter.

The data

A contingency table, also called a cross-tabulation, is a data table that displays the observed counts of categorical variables (Fagerland et al. 2017). These tables are often used to describe and analyze the relationship between two or more categorical variables (Fagerland et al. 2017).

Consider two categorical variables X_1 and X_2. Let V_1 and V_2 be their respective numbers of categories, and $\mathbf{X}$ the two-way contingency table associated to these two variables. The values of the first variable, say X_1, are represented by the rows of $\mathbf{X}$. The values of the second variable X_2 are represented by the columns of $\mathbf{X}$. The entries x_{ij} count the co-occurrences of the i^{th} value of the first variable and the j^{th} value of the second variable.

The contingency table $\mathbf{X}$ provides a summary of the variables. Thus, co-clustering $\mathbf{X}$ provides a summary of this summary.

Notations. This section uses the following notations.

N	number of observations, giving the total counts in $\mathbf{X}$.
V_1	number of values of the categorical variable X_1. It represents the number of rows of the contingency table $\mathbf{X}$.

V_2	number of values of the categorical variable X_2. It represents the number of columns of the contingency table $\mathbf{X}$.
i	index over the rows of $\mathbf{X}$, $1 \leq i \leq V_1$.
j	index over the columns of $\mathbf{X}$, $1 \leq j \leq V_2$.
g	the number of clusters of rows when it is specified.
m	the number of clusters of columns when it is specified.
k	index: $1 \leq k \leq g$.
l	index: $1 \leq l \leq m$.
G	number of co-clusters.
$\mathbf{z}$	matrix of affiliation of the rows to clusters of rows.
z_{ik}	an entry of $\mathbf{z}$.
$\mathbf{w}$	matrix of affiliation of the columns to clusters of columns.
w_{jl}	an entry of $\mathbf{w}$.
C_k^r	the k^{th} cluster of rows.
C_l^c	the l^{th} cluster of columns.
J_1	the number of clusters of rows when it is a model parameter.
J_2	the number of clusters of columns when it is a model parameter.
j_1	index: $1 \leq j_1 \leq J_1$.
j_2	index: $1 \leq j_2 \leq J_2$.
m_{j_1}	number of rows in the j_1^{th} cluster of rows.
m_{j_2}	number of columns in the j_2^{th} cluster of columns.
$N_{j_1 j_2}$	number of observations for the co-cluster formed by the j_1^{th} cluster of rows and the j_2^{th} cluster of columns.
$n_{i.}$	number of observations per row i.
$n_{.j}$	number of observations per column j.
$\mathcal{C}$	co-clustering criterion.

2.6.1 The Croki2 algorithm

Proposed by Govaert (1983), the Croki2 algorithm performs a co-clustering of contingency data. The algorithm tries to maximize the quantity of information included in the summary matrix $\hat{\mathbf{X}}$, measured by the χ^2 measure of information. Maximizing the information included in the summary matrix amounts to minimizing the loss of information due to regrouping the two sets into classes (Govaert and Nadif 2013).

The amount of information contained in the original contingency data $\mathbf{X}$ can be written as:

$$\chi^2(\mathbf{X}) = \sum_{i=1}^{V_1} \sum_{j=1}^{V_2} \frac{(x_{ij} - x_{i.}x_{.j})^2}{x_{i.}x_{.j}},$$

with $x_{i.}$ (respectively, $x_{.j}$) denoting the marginal row (respectively, column) frequencies. The amount of information in the summary matrix $\hat{\mathbf{X}}$ is:

$$\chi^2(\hat{\mathbf{X}}) = \sum_{k=1}^{g} \sum_{l=1}^{m} \frac{(\hat{x}_{kl} - \hat{x}_{k.}\hat{x}_{.l})^2}{\hat{x}_{k.}\hat{x}_{.l}},$$

with $\hat{x}_{k.}$ (respectively, $\hat{x}_{.l}$) denoting the marginal row (respectively, column) frequencies, and $\hat{x}_{kl}$ is the count per co-cluster $\hat{x}_{kl} = \sum_{i \in C_k^r, j \in C_l^c} x_{ij}$, where C_k^r and C_l^c denote the k^{th} cluster of rows and the l^{th} cluster of columns. g and m are the number of clusters of categories of the first variable (rows) and the number of clusters of categories of the second variable (columns), respectively.

The optimal partitions are those that minimize the difference between the two quantities $\chi^2(\mathbf{X})$ and $\chi^2(\hat{\mathbf{X}})$, thus minimizing the loss of information when representing or "replacing" $\mathbf{X}$ by the new contingency table $\hat{\mathbf{X}}$ or equivalently those maximizing the amount of information included in $\hat{\mathbf{X}}$. The algorithm iterates between finding a partition of rows and finding a partitions of columns until reaching an optimum for the co-clustering criterion $\chi^2(\hat{\mathbf{X}})$.

The Cemcroki2 alogirithm, proposed by Nadif and Govaert (2005) provides an extension of the Croki2 algorithm that, like Croki2, requires specifying the number of clusters in advance. In Nadif and Govaert (2005) and in Govaert and Nadif (2010), the models are adaptations of the latent block model for contingency tables, using Poisson distributions. The latent block models have been detailed in Section 2.5.3.

2.6.2 The Information-Theoretic co-clustering

Dhillon et al. (2003) proposed an information theoretic co-clustering algorithm for contingency tables. The approach views the contingency table as an empirical joint probability distribution of two discrete random variables that take values over the rows and columns and poses the co-clustering problem as an optimization problem in information theory. The optimal co-clustering is the one that maximizes the mutual information between the row clusters and column clusters or equivalently minimize the loss of mutual information between the original sets (the rows and columns) and the clustered sets (the clusters of rows and clusters of columns) under constraints on the number of row and column clusters.

The loss in mutual information is first written as Kullback-Leibler divergence between the original data distribution and an unknown distribution which can in turn be written as the product of the conditional distribution of rows given row clusters and the conditional distribution of columns given the column clusters. Then, the KL divergence is written as a weighted sum of the relative entropy between the row distribution and the row-cluster distribution or as a weighted sum of the relative entropy between the column distribution and the column-cluster distribution. For a fixed co-clustering:

$$\begin{aligned} \mathcal{C}(C_k^r, C_l^c) = KL(p(X_1, X_2)||q(X_1, X_2) &= \sum_{k,i|x_{i.} \in C_k^r} p(x_{i.})\, KL(p(X_2|x_{i.}) \,||\, q(X_2|C_k^r)) \\ &= \sum_{l,j|x_{.j} \in C_l^c} p(x_{.j})\, KL(p(X_1|x_{.j}) \,||\, q(X_1|C_l^c)), \end{aligned} \tag{2.4}$$

with X_1 and X_2 are the random variables representing the set of rows and columns, and $q(x_{i.}, x_{.j}) = p(z_{ik} = 1, w_{jl} = 1)p(x_{i.}|z_{ik} = 1)p(x_{.j}|w_{jl} = 1)$ for $x_{i.} \in C_k^r$ and $x_{.j} \in C_l^c$.

The algorithm starts with some initial random partition and iteratively computes marginals $p(x_{i.})$ and $p(x_{.j})$. It computes the row-cluster prototypes $q(x_{.j}|C_k^r) = q(x_{.j}|C_l^v)q(C_k^r|C_l^c)$ and the column cluster prototypes $q(x_{i.}|C_l^c) = q(x_{i.}|C_k^r)q(C_k^r|C_l^c)$, then assigns each row to its closest row-cluster prototype minimizing the divergence between $q(X_1|C_l^c)$ and $p(X_1|x_{.j})$ and each column to its closest column-cluster prototype minimizing the divergence between $q(X_2|C_k^r)$ and $p(X_2|x_{i.})$. The algorithm monotonically increases the preserved mutual information and eventually converges to a local minimum. However, the approach is restricted to non-negative count data and requires specifying the number of row and column clusters.

A similar co-clustering approach that is based on entropy regularized optimal transport has been proposed recently by Laclau et al. (2017). The solution of the optimal transportation problem is obtained from a doubly stochastic coupling matrix representing an approximation of the joint probability distribution of the original data. The coupling matrix is factorized into three terms where one of them reflects the posterior distribution of data given co-clusters while the two others represent the approximated distributions of the rows and columns. The approximated distributions are then used to obtain the final row and column partitions. Similarly to (Dhillon et al. (2003)), the approach looks for a factorization of the joint probability distribution between two variables X_1 and X_2 which is estimated from the data matrix. However, while Dhillon et al. (2003) minimize the Kullback-Leibler divergence $KL(p(X_1, X_2)|q(X_1, X_2))$, the approach proposed by Laclau et al. (2017) minimizes $KL(q(X_1, X_2)|p(X, X_2))$.

2.6.3 The MODL approach

The MODL approach (Boullé 2007) is a non-parametric model selection based approach for conditional and joint density estimation. In the considered models, named *data grid models*, each variable is partitioned in intervals or groups of values according to whether it is numerical or categorical. The whole data is then partitioned into a grid of cells resulting from the cross-product of the variable partitions. The model selection is based on the maximization of the likelihood of the data, penalized by a prior related term. The model parameters form a hierarchy where each parameter is chosen according to the previous ones. Consequently, the prior distribution is itself hierarchical. At each level of the hierarchy, the parameter distribution is considered flat, which introduces the least a priori knowledge about the data. Once the data is observed, the penalized likelihood evaluates the posterior probability of the parameters given the observed data.

For the purpose of this work, we will focus on the application of MODL approach on estimating the joint density between two categorical variables

X_1 and X_2. For more details about the other applications, readers are refered to the complete presentation of the approach in Boullé (2007).

Let V_1 and V_2 be the number of categories of the variables X_1 and X_2, and N the number of observations (sum of the contingency table formed by X_1 and X_2). The grid nature of the approach performs a compression of the values of the variables with respect to the number of times these values have been observed together. The parameters of a MODL model are of the form:

- The number of groups per variable J_1 and J_2. The number of groups is chosen with equal probabilities from 1 to the number of unique values of the variable (V_1 or V_2). These numbers of groups define the size of the data grid $G = J_1 \times J_2$.

- The partition of the values of a variable into the previously chosen number of groups, resulting in the counts $\{m_{j_1}\}_{1 \leq j_1 \leq J_1}$ and $\{m_{j_2}\}_{1 \leq j_2 \leq J_2}$ of the number of categorical values per group. All possible partitions are equally probable.

- The distribution of the N observations into the cells of the grid, resulting in the counts $\{N_{j_1 j_2}\}_{1 \leq j_1 \leq J_1, 1 \leq j_2 \leq J_2}$ per cell. All possible distributions are equally probable. The number of observations per group (marginal counts of the summary matrix) can now be deduced by summation:
$$N_{j_1 \cdot} = \sum_{j_2} N_{j_1 j_2} \text{ and } N_{\cdot j_2} = \sum_{j_1} N_{j_1 j_2}.$$

- For each group of values, the distribution of the $N_{j_1 \cdot}$ (resp. $N_{\cdot j_2}$) observations of the group on the m_{j_1} (resp. m_{j_2}) values in the group, resulting in the counts $n_{i \cdot}$ (resp. $n_{\cdot j}$), giving the number of observations per value i (resp. j). All possible distributions of the values on the groups are equally probable.

At each level of the hierarchy, the parameters are chosen conditionally on the previously chosen ones. However, within the same level, the parameters are considered independent. With this in mind, the prior probability of the model parameters M can be written:

$$\begin{aligned} P(M) = & P(J_1|V_1)P(J_2|V_2) \\ & P(\{m_{j_1}\}|J_1)P(\{m_{j_2}\}|J_2) \\ & P(\{N_{j_1 j_2}\}|N, J_1, J_2) \\ & P(\{\{n_{i \cdot}\}\}|\{N_{j_1 j_2}\}, J_1, J_2) \\ & P(\{\{n_{\cdot j}\}\}|\{N_{j_1 j_2}\}, J_1, J_2). \end{aligned}$$

Following the hierarchy of the parameters, the respective probabilities are as follows:

- the choice of the number of groups per variable is governed by $P(J_1|V_1)$ and $P(J_2|V_2)$ given by:
$$P(J_1|V_1) = \frac{1}{V_1} \text{ and } P(J_2|V_2) = \frac{1}{V_2},$$

- the choice of the partition of the V_1 (resp. V_2) values into J_1 (resp. J_2) groups is governed by $P(\{m_{j_1}\}|V_1)$ (resp. $P(\{m_{j_2}\}|V_2)$) given by:

$$P(\{m_{j_1}\}|J_1) = \frac{1}{B(V_1, J_1)} \text{ and } P(\{m_{j_2}\}|J_2)) = \frac{1}{B(V_2, J_2)},$$

where $B(V, J)$ is the sum of Stirling numbers of the second kind $B(V, J) = \sum_{j=1}^{J} S(V, j)$, giving the number of possible ways of partitioning V values into J groups.

- the choice of a distribution of the N observations into the $G = J_1 \times J_2$ cells of the grid is governed by $P(\{N_{j_1 j_2}\}|N, J_1, J_2)$, where $N_{j_1 j_2}$ is the number of observations to be associated to the grid formed by the j_1^{th} group of the first variable and the j_2^{th} group of the second variable. The probability of choosing such a distribution is given by:

$$P(\{N_{j_1 j_2}\}|N, J_1, J_2) = \frac{1}{\binom{N+G-1}{G-1}},$$

- the choice of a distribution of the $N_{j_1\cdot}$ (resp. $N_{\cdot j_2}$) observations in a group into the m_{j_1} values of the group is governed by $P(\{n_{i\cdot}\}|\{N_{j_1 j_2}\}, J_1, J_2)$ (resp. $P(\{n_{\cdot j}\}|\{N_{j_1 j_2}\}, J_1, J_2)$), where $n_{i\cdot}$ is the number of observations to be associated to the value i of the group j_1 and $n_{\cdot j}$ is the number of observations to be associated to the value j of the group j_2 of the second variable. For a single group of values, the probability of choosing such a distribution is given by:

$$P(\{n_{i\cdot}\}|\{N_{j_1 j_2}\}, J_1, J_2) = \frac{1}{\binom{N_{j_1\cdot}+m_{j_1}-1}{m_{j_1}-1}}, \text{ and}$$
$$P(\{n_{\cdot j}\}|\{N_{j_1 j_2}\}, J_1, J_2) = \frac{1}{\binom{N_{\cdot j_2}+m_{j_2}-1}{m_{j_2}-1}}.$$

Under the condition of independence between the parameters within the same level of the hierarchy, the probability of choosing these distributions simultaneously for all groups is given by:

$$P(\{\{n_{i\cdot}\}\}|N, J_1, J_2) = \prod_{j_1=1}^{J_1} P(\{n_{i\cdot}\}|N, J_1, J_2) = \prod_{j_1=1}^{J_1} \frac{1}{\binom{N_{j_1\cdot}+m_{j_1}-1}{m_{j_1}-1}}, \text{ and}$$
$$P(\{\{n_{\cdot j}\}\}|N, J_1, J_2) = \prod_{j_2=1}^{J_2} P(\{n_{\cdot j}\}|N, J_1, J_2) = \prod_{j_2=1}^{J_2} \frac{1}{\binom{N_{\cdot j_2}+m_{j_2}-1}{m_{j_2}-1}}.$$

Given the model parameters, the likelihood of the data is defined by the likelihood of the multinomial distribution of the observations on the data grid cells and the multinomial distribution of the observations per group over the values in the group:

- the likelihood of the multinomial distribution of the observations on the data grid cells, with the counts $\{N_{j_1 j_2}\}$ is given by

$$\frac{\prod_{j_1=1}^{J_1} \prod_{j_2=1}^{J_2} N_{j_1 j_2}!}{N!},$$

- the likelihood of the multinomial distribution of the observations in a group over the values in the group are given by

$$\frac{\prod_{i \in j_1} n_{i.}!}{N_{j_1.}!} \text{ and } \frac{\prod_{j \in j_2} n_{.j}!}{N_{.j_2}!},$$

where $i \in j_1$ (resp. $j \in j_2$) means that the value i (resp. j) belongs to the j_1^{th} (resp. j_2^{th}) group of the corresponding variable.

Hence, the multinomial distribution of the marginal observations per variable over the values of the variable are given by:

$$\frac{\prod_{i=1}^{V_1} n_{i.}!}{\prod_{j_1=1}^{J_1} N_{j_1.}!} \text{ and } \frac{\prod_{j=1}^{V_2} n_{.j}!}{\prod_{j_2=1}^{J_2} N_{.j_2}!}.$$

The likelihood of the full set of parameters M is defined by the product of the likelihoods of the individual parameters. In order to select the best set of parameters, a MAP based criterion is optimized to maximize their probability given the data $P(M|D) \propto P(M)P(D|M)$ as shown by Theorem 1. The prior distribution on the model parameters serves as a regularization term which prevents the optimization from selecting systematically a high number of groups and prevents over-fitting.

Theorem 1 *A MODL co-clustering is Bayes optimal if the evaluation of its parameters according to the following criterion is minimal (Boullé (2011)):*

$$\begin{aligned}
\mathcal{C}(M) =& -\log P(M) - \log P(D|M) \\
=& \log V_1 + \log V_2 + \log B(V_1, J_1) + \log B(V_2, J_2) \\
&+ \log \binom{N+G-1}{G-1} + \sum_{j_1=1}^{J_1} \log \binom{N_{j_1.}+m_{j_1}-1}{m_{j_1}-1} + \sum_{j_2=1}^{J_2} \log \binom{N_{.j_2}+m_{j_2}-1}{m_{j_2}-1} \\
&+ \log N! - \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \log N_{j_1 j_2}! + \sum_{j_1=1}^{J_1} \log N_{j_1.}! + \sum_{j_2=1}^{J_2} \log N_{.j_2}! \\
&- \sum_{i=1}^{V_1} \log n_{i.}! - \sum_{j=1}^{V_2} \log n_{.j}!.
\end{aligned} \tag{2.5}$$

The first line of this criterion corresponds to the prior distribution of choosing the numbers of groups and to the partition of the values of each

variable to the chosen number of groups. The second line represents the specification of the parameters of the multinomial distribution of the N observations on the G cells of the data grid and the specification of the multinomial distribution of the observations of each group on the values of the group. The third line corresponds to the likelihood of the distribution of the observations on the data grid cells and the likelihood of the distribution of the observations per group over the values in the group, by the mean of a multinomial term.

The approach presented so far performs a simultaneous clustering of the values of two variables ($X_1 \times X_2$) by the medium of partition of the observed counts on a data grid. In the next chapter, we show how this co-clustering approach can be applied to the problem of co-clustering instances$\times$variables, in the presence of mixed type variables.

2.7 Summary

In this chapter, we have explored the most commonly known pattern extraction and exploratory analysis techniques. Namely, dimensionality reduction, clustering techniques, and co-clustering, with a particular emphasis on the co-clustering based techniques. Through this chapter, we have seen that each of these techniques is adapted to a certain type of data. The only co-clustering like method that handles mixed-type variables is Crosscat (Mansinghka et al. 2016) which, in the classical sense, is not really a co-clustering technique since it provides one clustering of the variables and multiple clusterings of the set of instances.

In an attempt to solve this problem of mixed-type data co-clustering, we have extended the latent block models to handle data containing both numerical and binary variables (see Appendix B). While this extension enhances the performance of the co-clustering when the clusters are intrinsically defined by mixed variables, it still suffers from the same problem as most of the above mentioned co-clustering methods, which is the need to have the numbers of clusters specified in advance. For latent block models, model selection criteria like the Bayesian Information Criterion BIC (Schwarz 1978), the Akaike Information Criterion AIC (Akaike 1974), and the Integrated Classification Criterion ICL (Biernacki et al. 2000) can be used to infer the optimal number of clusters but the proposed extension does not extend these criteria for mixed data. Furthermore, this extension inherits the inapplicability on large data sets from the underlying LBM model. To handle these problems, we propose using the MODL approach to perform a co-clustering of data containing mixed variables. The first advantage for using MODL is that it extracts the inter-dependencies between the variables in a non parametric manner. Therefore, there is no need to know the numbers of clusters in advance. A second advantage is that it scales and can be applied to large data sets while still providing easily interpretable clusters, which is an important advantage in exploratory analysis. This approach is discussed in details in the next chapter.

Part II

Contributions

Chapter 3

Co-clustering mixed data

In this chapter, we introduce a new MODL based methodology to perform a co-clustering of the instances and variables of a data table. The objective of the approach is to find a co-clustering structure that expresses the interdependence between variables of possibly mixed types and provide easily interpretable clusters even when the data set is large and complex.

This chapter is organized as follows. In order to set the motivating background for the proposed co-clustering approach, Section 3.1 introduces a close-up emphasis on the advantages of value based co-clustering compared to the full-row and full-column based techniques. Section 3.2 presents the co-clustering approach. We compare the results of the proposed approach with those of the Multiple Correspondence Analysis approach, presented in Section 3.4. The experimental results are provided in Section 3.5. In Section 3.6, we conclude and summarize the main limitations of the approach, which sets the motivation for the next chapter.

3.1 Introduction

Clustering methods cluster objects (instances) based on their observed values with respect to all the studied variables. By contrast, as discussed in the previous chapter, co-clustering methods try to extract a structure of homogeneous blocks from data. This extraction can be performed via a mapping from the set of instances to a set of clusters of instances and a mapping from the set variables to the clusters of variables, or through a direct mapping from the set of entries in the data matrix to clusters of entries. In the first case, the entries at the intersection of a cluster of instances and a cluster of variables form a co-cluster while in the latter, the clusters of entries are themselves co-clusters. In the traditional co-clustering methods (defined by two mappings), a cluster of instances contains instances that behave similarly with respect to all the variables but with varying degrees (as defined by the blocks), and similarly, the variables within a cluster are observed similarly with all the objects but with varying degrees (as defined by the blocks). Examples of these techniques include those fetching for the structures in Figures 2.1b and 2.1c (see Section 2.4.3 of the previous chapter).

However, it is often the case that the instances do not exhibit similar patterns in all variables but in a subset of variables, and inversely. This is the main motivation behind the second type of co-clustering methods (defined by a mapping of the data entries). This point of view has been particularly exploited in gene expression bi-clustering techniques where most methods try to identify genes (instances) that have correlated expression values in various conditions (variables), known as co-expressed genes (Madeira and Oliveira 2004). Most gene expression bi-clustering methods are used to capture the genes that are correlated only for a subset of conditions, which is biologically interesting since not only it allows to capture the correlated genes, but also enables the identification of genes that do not behave similarly in all conditions (Eren et al. 2013). Examples of these techniques are those fetching for the structures in Figures 2.1d, 2.1e, and 2.1i (see Section 2.4.3).

In this chapter, we adopt this point of view and we go a step further to argue that the instances within a cluster of instances (in particular) need neither to be similar over all the variables (as in the traditional view) nor to be similar over a subset of variables (as in the co-expressed-genes view) but they need only to be similar with respect to a subset of values of the variables, and with varying degrees (as defined by the blocks). Inversely, the variables do not need to be similar with respect to all the instances. In particular, two variables can be partly correlated. This partial correlation can not be explained by a clustering that is based on all of their observed values. To this end, we propose a co-clustering approach that is based on MODL (Section 2.6.3) which is the subject of this chapter.

3.2 The co-clustering approach

Let $\mathbf{X}$ be a data table containing I instances and K variables. Let the set of variables be denoted $\mathcal{X}$. Among these variables, suppose K_n are numerical and form the set $\mathcal{X}_n$. The remaining K_c variables are categorical and form the set $\mathcal{X}_c$. Each entry x_{ij}, numerical or categorical, is called *observation*. The observation $x_{ij} = v$ says that the value of variable X_j is v for object i.

We propose to form a co-clustering method of the data table $\mathbf{X}$ using a two-step methodology. The first step consists in homogenizing the variables, then transforming the data into two categorical variables that capture the relationship between the instances and the homogenized variables. The second step consists of applying a standard co-clustering approach to the transformed homogeneous data. Our objective is to require no model related parameters, such as the number of clusters. Therefore, in the co-clustering step, we will use the MODL approach (Boullé 2011) for its non parametric nature, its efficiency for extracting correlation structures, its scalability and its robustness to over-fitting, induced by the embedded regularization. This section presents this two-step approach in more details.

3.2.1 Variable parts

For the double objective of simultaneously processing variables of different types and capturing local as well as global correlations between the variables, some homogenizing technique is required. For this purpose, we introduce the notion of *variable parts*. Partitioning the variables enables easier approximation of the joint densities. For example, given two variables, the idea is that if they are partitioned (the distribution of every variable is approximated by intervals or groups of values), then if the variables are correlated, their corresponding partitions will be correlated. If the variables are partly or locally correlated, then some parts the variables will be correlated, and if they are uncorrelated, then their partitions will still be uncorrelated.

Naturally, the partitioning of the variables should approximate their distribution as accurately as possible. Nevertheless, to avoid over-fitting, the partitions should not be too accurate. Hence, the choice of the partitions should take into account this trade off between fitness to data and complexity of the resulting co-clustering.

3.2.2 Creating variable parts: data pre-processing

For the homogenizing step, we choose partitioning all variables using a user parameter p, which represents the maximal number of parts per variable. One possible discretization method is with equal-width but we choose to leverage a frequency based discretization because it reinforces the robustness of the approach and minimizes the effect of outliers, if present in the data. In particular, the values of a numerical variable are transformed into p contiguous intervals while the values of the categorical variables are transformed into p groups of values as follows:

- in the case of a numerical variable, the parts are the result of an unsupervised discretization of the range of the variable into p intervals with equal frequencies,
- in the case of a categorical variable, we choose to use the $p-1$ most frequent values to define the first $p-1$ parts, and to put the other less frequent values into the p^{th} part.

We use the term *part* to denote both a numerical interval and a categorical group.

On the choice of the dicretization parameter

The dicretization parameter p defines the maximal granularity at which the analysis can be performed. A good choice of p is related to a trade-off between the fineness of the analysis, the time required to compute the co-clustering of the second step, and the interpretability of the co-clustering results.

In theory, the computational cost of the MODL co-clustering ($\mathcal{O}(N\sqrt{N}\log N)$ where $N = I \times K$ is the total number of observation

(Section 3.2.3)) does not depend on the parameter p, but in practice, the observed computation time tends to decrease with smaller values of p. Also, the size of the data set and its complexity can be taken as an indicator. Small values are probably sufficient for small and simple data sets while for larger ones, it would be wise to choose a larger parameter p. However, we would recommend to start with high values of p since it gives a detailed description of the data, which minimizes the loss of information.

On the Iris data set for example, we choose $p = 5$ for this dicretization step. The results of this dicretization are illustrated in Table 3.1.

SepalLength	*SepalWidth*	*PetalLength*	*PetalWidth*	*Class*
$]-\infty; 5.05]$	$]-\infty; 2.75]$	$]-\infty; 1.55]$	$]-\infty; 0.25]$	$\{setosa\}$
$]5.05; 5.65]$	$]2.75; 3.05]$	$]1.55; 3.95]$	$]0.25; 1.15]$	$\{versicolor\}$
$]5.65; 6.15]$	$]3.05; 3.15]$	$]3.95; 4.65]$	$]1.15; 1.55]$	$\{virginica\}$
$]6.15; 6.55]$	$]3.15; 3.45]$	$]4.65; 5.35]$	$]1.55; 1.95]$	
$]6.55; +\infty[$	$]3.45; +\infty[$	$]5.35; +\infty[$	$]1.95; +\infty[$	

Table 3.1 – *The output of the discretization step on iris, for $p = 5$.*

In Table 3.1, the numerical parts are represented as intervals because they contain contiguous values with boundaries referring to the boundaries at which the cuts are made. The categorical parts are represented by their constituting categorical values. This representation is useful for later interpretation of the clusters.

3.2.3 Data transformation

Although designed for joint density estimation, MODL has also been applied to the case of instances×binary-variables. An example of such application is that of a large corpus of documents, where each document is characterized by tens of thousands of binary variables representing the usage of words. In this case, the corpus of documents is transformed beforehand into a representation in the form of two variables *IdText* and *IdWord*. In the same manner, we transform the discretized data resulting from the pre-processing step into two categorical variables *IdInstance* and *IdVarPart* in order to use the MODL approach for co-clustering these two categorical variables (as detailed in Section 2.6.3).

The discretized data is transformed into two variables *IdInstance* and *IdVarPart* by creating, for each instance, a record per variable that logs the link between the instance and its variable part. The set of I initial instances characterized by K variables is thus transformed into a data set of $N = I \times K$ new instances and two new categorical variables, the first of which contains $V_1 = I$ values and the second contains, at most, $V_2 = K \times p$ values. The approach detailed in Section 2.6.3 can now be applied directly to these new categorical variables.

For example, for the Iris data set, this transformation results in two columns of 750 instances. Table 3.2 shows the first ten instances.

IdInstance	*IdVarPart*
$I1$	$SepalLength]5.05; 5.65]$
$I1$	$SepalWidth]3.45; +\infty[$
$I1$	$PetalLength] - \infty; 1.55]$
$I1$	$PetalWidth] - \infty; 0.25]$
$I1$	$Class\{setosa\}$
$I2$	$SepalLength] - \infty; 5.05]$
$I2$	$SepalWidth]2.75; 3.05]$
$I2$	$PetalLength] - \infty; 1.55]$
$I2$	$PetalWidth] - \infty; 0.25]$
$I2$	$Class\{setosa\}$

Table 3.2 – *The first 10 instances of the transformed Iris data.*

Note that after the pre-processing step, the co-clustering can not leverage two aspects of the data: the actual value taken by a variable inside a variable part and the original links between variable parts. In other words, the approach is oblivious to the fact that $SepalLength]5.05; 5.65]$ and $SepalLength] - \infty; 5.05]$ both refer to the same original variable and to the fact that they are contiguous (these parts are now two distinct categorical values). In the formalized model, presented in Chapter 4, we will take into account this link between the variable parts and we will show that it proves to be useful.

3.2.4 The co-clustering

Now that our data is represented in the form of two categorical variables, we can apply MODL to estimate the joint probability distribution between these two variables. This results in two partitions of the values of the newly introduced categorical variables. Clusters of values of the variable *IdInstance* are clusters of instances while clusters of values of the variable *IdVarPart* are clusters of variable parts. Thus, the results consists in a form of co-clustering in which variables are clustered at the level of parts rather than globally. As a result, the co-clustering consists of forming clusters of instances and clusters of variable parts, which has the advantage of enabling mixed clusters of variables. In the resulting co-clustering, the instances of the original data set (values of the variable *IdInstance*) are grouped if they are distributed similarly over the groups of variables parts (values of the variable *IdVarPart*), and vice-versa. Also, the number of clusters does not need to be specified because MODL optimizes the number of groups of values of the clustered variables.

3.3 Exploratory analysis of the results

Given a co-clustering, defined by its set of parameters (see Section 2.6.3), denote:

- G_u, the number of clusters of instances;
- G_p, the number of clusters of variable parts;
- $\{m_i\}_{1\leq i\leq G_u}$, the number of instances in the i^{th} cluster of instances;
- $\{m_j\}_{1\leq j\leq G_p}$, the number of parts in the j^{th} cluster of variable parts;
- $\{N_{i,j}\}_{1\leq i\leq G_u, 1\leq j\leq G_p}$, the number of observations in the co-cluster (C_k^u, C_j^p) formed by the k^{th} cluster of instances and the j^{th} cluster of parts;
- $N_{i.} = \sum_j N_{i,j}$, the numbers of observations in cluster of instances C_k^u;
- $N_{.j} = \sum_i N_{i,j}$, the numbers of observations in cluster of variable parts C_j^p.

To facilitate the analysis of the results, we use model coarsening to simplify the co-clustering and mutual information to explain the clusters.

3.3.1 Model coarsening

When the optimal co-clustering is too detailed (the matrix of co-clusters is large), coarsening of the partitions (as proposed in Guigourès et al. (2015a)) can be implemented by merging clusters (of objects or variable parts) in order to obtain a simplified structure. While this model coarsening approach can degrade the co-clustering quality, the induced simplification enables the analyst to gain insight on complex data at a coarser level, in a way similar to exploration strategies based on hierarchical clustering. The dimension on which the merging is performed and the best merging are chosen optimally at each coarsening step with regards to the minimum divergence from the optimal co-clustering, measured by the difference between the optimal value of the MODL criterion (Section 2.6.3) and the value obtained after merging the clusters (in the next chapter (Chapter 4), we will propose a new criterion).

3.3.2 The mutual information between clusters

Given a desired coarsening level, the result of the co-clustering approach is represented as a matrix of counts of the number of observations per co-cluster. This can be seen as a contingency table between the clusters of instances and the clusters of variable parts. Each co-cluster associates the variable parts that constitute the cluster to one another and associates these

variable parts to the clusters of instances with varying degrees, depending on their contribution to the cluster of instances. Inversely, each cluster of instances is explained by the clusters of variable parts with varying degrees. To measure this association between the clusters, we compute the matrix of mutual information.

The mutual information $I = \{I_{(C_k^u, C_l^p)}\}$ measures the divergence from the independence. An entry $I_{(C_k^u, C_l^p)}$ is given by:

$$I_{(C_k^u, C_l^p)} = P(C_k^u, C_l^p) \log \frac{P(C_k^u, C_l^p)}{P(C_k^u) \times P(C_l^p)},$$

where $P(C_k^u, C_l^p)$, $P(C_k^u)$ and $P(C_l^p)$ are the frequency of a co-cluster, the frequency of a cluster of instances and the frequency of a cluster of variable parts, which are given respectively by:

$$P(C_k^u, C_l^p) = \frac{N_{i,j}}{N},\ P(C_k^u) = \frac{N_{i.}}{N}, \text{ and } P(C_l^p) = \frac{N_{.j}}{N}.$$

This matrix of mutual information is used for basis in associating clusters of one type to the clusters of the second type with respect to the contribution of their mutual information to the total information. For visualization, we use a color-coded version of this matrix where, in each cell, red colors represent an over-representation of the instances compared to the case where the two dimensions are independent and blue colors represent an under-representation (see Figure 3.1 for example). White cells represent empty co-clusters (no association between the corresponding clusters). A cluster of instances is described by the parts contained in the cluster(s) of variable parts with the highest contribution to the total mutual information of the cluster (of instances).

3.3.3 Summary

In summary, the proposed co-clustering approach takes a mixed-type data matrix as input and performs a clustering of the instances and a clustering of the variables at the level of variable parts. Consequently, values coming from different variable types are grouped within the same co-clusters. However, the approach requires a user parameter for discretization. Nevertheless, one should note that this granularity parameter is far less restrictive than other common parameters such as the number of instance clusters and the number of variable clusters, commonly required by the vast majority of co-clustering methods. In the formalized model (Chapter 4), this parameter will be optimized within the model but for the remaining of this chapter we will choose it manually for each data set.

Since we are in the context of exploratory analysis of a mixed-type data table, we compared our methodology to the most widely used factor analysis method in case of the presence of categorical variables: Multiple Correspondence Analysis (MCA). MCA is chosen as a comparison basis because it enables extracting the correlations between categorical variables while performing a clustering of the instances. Note, however, that this method does

not take the original mixed-type data set as input. Therefore, we will compare the co-clustering part of the approach to the results of MCA. That is, we will apply MCA on the discretized data, resulting from the pre-processing step. Details of the Multiple Correspondence Analysis techniques, and how it can be related to the singular value decomposition based co-clustering, are addressed below.

3.4 Multiple Correspondence Analysis

Factor analysis is a set of statistical methods, the purpose of which is to analyze the relationships or associations that exist in a data table, where rows represent instances and columns represent variables (of any type). The main purpose of these methods is to determine the level of similarity (or dissimilarity) between groups of instances (problem classically treated by clustering) and the level of associations (correlations) between the observed variables. Multiple Correspondence Analysis (MCA) is a factor analysis technique that enables one to analyze the dependencies between multiple categorical variables while performing a typology (grouping) of instances and variables in a complementary manner. We argue that these goals are very close to those of co-clustering and that by placing MCA in a spectral clustering context, one can see the similarity between the use of the singular value decomposition in co-clustering and its use in MCA.

MCA in practice

MCA applies to categorical data. Let $\mathbf{Y} = (y_{ik_c})_{1 \leq i \leq I, 1 \leq k_c \leq K_c}$ be the table of I instances described by K_c categorical variables, and let m_{k_c} be the number of categories for the k_c^{th} variable. MCA uses an $I \times m$ indicator matrix $\mathbf{Z}$ (with $m = \sum_{k_c=1}^{K_c} m_{k_c}$) called complete disjunctive table (CDT). This CDT is a juxtaposition of K_c indicator matrices of all variables where rows represent the instances and columns represent the categories of the variable, and such that $\mathbf{Z}_{ij} = 1$ iff the i^{th} instance takes the category j for a given variable. To put it simply, the matrix $\mathbf{Z}$ can be considered as a contingency table between the instances and the set of all categories in the data.

The most common way to perform MCA is by applying a correspondence analysis algorithm to the indicator matrix $\mathbf{Z}$ and/or to the Burt matrix $B = \mathbf{Z}^T\mathbf{Z}$ (^{T} is the transpose), but one equivalent approach is to perform a SVD on a standardized matrix.

3.4.1 MCA as a correspondence analysis method

The CDT $\mathbf{Z}$ has some known characteristics that makes it easily exploitable. For instance, the sum of all elements of each row is equal to the number K_c of variables, the sum of all elements of a column j is equal to the marginal

frequency n_j of the corresponding category, the sum of all columns in each indicator matrix is equal to 1, the sum of all elements in $\mathbf{Z}$ is equal to $I \times K_c$. Also, assuming equally important instances, the matrix of instance weights is given by $r = \frac{1}{I}\mathbf{1}$ ($\mathbf{1}$ is the identity matrix), and the column weights are given by the diagonal matrix $D = diag(D_1, D_2, \ldots, D_{K_c})$ where each D_{k_c} is the diagonal matrix containing the marginal frequencies of all categories of the k_c^{th} variable.

The results that we will exploit from an MCA are the projection of the instances and categories on the principal axis, which are given as follows.

1. The principal coordinates of categories are given by the eigenvectors of $\frac{1}{K_c}\mathbf{D}^{-1}\mathbf{Z}^T\mathbf{Z}$, which are the solutions of the equation:

$$\frac{1}{K_c}\mathbf{D}^{-1}\mathbf{Z}^T\mathbf{Z}\mathbf{a} = \mu\mathbf{a}.$$

2. The principal coordinates of instances are given by the eigenvectors of $\frac{1}{K_c}\mathbf{Z}\mathbf{D}^{-1}\mathbf{Z}^T$, which are the solutions of the equation:

$$\frac{1}{K_c}\mathbf{Z}\mathbf{D}^{-1}\mathbf{Z}^t\mathbf{z} = \mu\mathbf{z}.$$

3. The transition formulas (Saporta (2006)) are given by $\mathbf{z} = \frac{1}{\sqrt{\mu}}\frac{1}{K_c}\mathbf{Z}\mathbf{a}$ and $\mathbf{a} = \frac{1}{\sqrt{\mu}}\mathbf{D}^{-1}\mathbf{Z}^T\mathbf{z}$, which describes how to pass between the coordinates.

Additional information that might aid in the interpretation of the results come from the following.

- The total inertia is equal to $(\frac{m}{K_c} - 1)$.
- The inertia of all the m_{k_c} categories in the k_c^{th} variable is equal to $\frac{1}{K_c}(m_{k_c} - 1)$. Since the contribution of a variable to the total inertia is proportional to the number of categories in the variable, it is preferable to require all variables to have roughly the same number of categories. Hence, another utility of the pre-processing step.
- The contributions of an instance i and of a category j to a principal axis h are given by:

$$Ctr_h(i) = \frac{1}{I}\frac{z_{ih}^2}{\mu_h} \text{ and } Ctr_h(j) = \frac{n_j}{IK_c}\frac{a_{jh}^2}{\mu_h}.$$

- The contribution of a variable to the inertia of a factor is equal to the sum of contributions of all categories in the variable to that same axis. This contribution measures the level of correlation between the variable and the principal axis.

3.4.2 MCA as a spectral technique

One can think of the aims of the MCA as equivalent to those of co-clustering because when placing MCA in a spectral context, one can see the similarity between the use of spectral techniques in co-clustering and its use in MCA. Let $\mathbf{P} = \frac{1}{I \times K_c}\mathbf{Z}$ be the correspondence matrix (the sum of the entries in $\mathbf{Z}$ is $I \times K_c$), r the vector of row sums and c the vector of column sums of $\mathbf{P}$ ($r_i = \sum_{k_c=1}^{K_c} P_{ik_c}$ and $c_{k_c} = \sum_{i=1}^{I} P_{ik_c}$). MCA can be performed using a SVD of a standardized matrix derived from the correspondence matrix. For instance, when working with the indicator matrix, MCA requires a SVD of

$$S_1 = D_r^{-1/2}\mathbf{P}D_c^{-1/2}. \tag{3.1}$$

When using the Burt matrix, MCA requires a SVD on the standardized residual matrix:

$$S_2 = D^{-1/2}(\mathbf{F} - \tilde{r}\tilde{r}^T)D^{-1/2},$$

where $\mathbf{F} = \frac{B}{I \times K_c^2}$ is the correspondence matrix derived from the Burt table B, and $\tilde{r}$ is the vector of row sums of $\mathbf{F}$ (or column sums since $\mathbf{F}$ is symmetric), and $D = D_{\tilde{r}}$ is a diagonal matrix containing the elements of $\tilde{r}$ (see D'Enza and Greenacre (2012) and Di Ciaccio et al. (2012)).

Recall that singular value decomposition SVD is a linear algebra technique that expresses an $n \times m$ matrix A as the product $A = U\mathbb{A}V^T$ where $\mathbb{A}$ is a diagonal matrix with non negative entries λ_i (called singular values) which are the square roots of the eigenvalues of AA^T and U and V are $n \times min(n, m)$ and $m \times min(n, m)$ orthogonal column matrices. The columns of U, called left singular vectors of A, are the eigenvectors of AA^T and the columns of V, called right singular vectors of A, are the eigenvectors of A^TA.

From the SVD of S_1, we get the spectral decomposition:

$$S_1 = U\mathbf{\Sigma}_1V^T,$$

where U and V are the matrices of right and left singular vectors with constraints $UU^T = \mathbf{1}$ and $V^TV = \mathbf{1}$ ($\mathbf{1}$ is the identity matrix) and $\mathbf{\Sigma}_1$ is the diagonal matrix of nonnegative singular values.

The standard coordinates for rows and columns are given, respectively, by:

$$R_s = D_r^{-1/2}U \text{ and } C_s = D_c^{-1/2}V.$$

The principal coordinates of rows and columns are given, respectively, by:

$$R_p = D_r^{-1/2}U\mathbf{\Sigma}_1 \text{ and } C_p = D_c^{-1/2}V\mathbf{\Sigma}_1.$$

From the SVD of S_2, we get the spectral decomposition:

$$S_2 = W\mathbf{\Sigma}_2W^T,$$

where $\mathbf{\Sigma}_2$ is the diagonal matrix of singular values in descending order and W is the matrix of singular vectors.

The principal coordinates of the rows, or columns (since S_2 is symmetric) are

$$F = D^{-1/2}W\Sigma.$$

When applying a clustering technique to the principal coordinates of the rows and columns, a structure of correspondence between the categories and instances emerges. For example, this is the case in the approach used in Dhillon (2001) which is based on dimentionality reduction and a clustering algorithm (see Section 2.5.2). In Dhillon (2001), the partitioned data contains both row projections and column projections on the second eigenvector which usually expresses the strongest associations and the best seperation between the clusters. However, Kluger et al. (2003) observe that, while the partitioning eigenvectors (those providing the best separation) are commonly associated with the second largest eigenvalue, it can also be one of the eigenvectors associated to one of the following largest values and, in some cases, the partitioning eigenvector can be one of the eigenvectors associated to a small eigenvalue. Hence, a prudent strategy is to examine all the eigenvectors. In Kluger et al. (2003), the authors do exactly this as they perform a partitionning of each eigenvector and choose the best candidate as the one that can be best approximated by a piece wise constant vector. To this end, they examine all possible partitions of each ordered vector into a number of parts, for all possible numbers of parts.

3.4.3 Summary

Although multiple correspondence analysis and co-clustering are different in some aspects, we have seen that they both have the same broader goal of discovering the correspondence or association between the objects and features and when they are looked at from a spectral clustering angle, we hope to make a rightful comparison. In the following we will compare the proposed co-clustering approach to performing a clustering using all the eigenvectors when possible or the ones associated to the highest eigenvalues otherwise.

3.5 Experiments and comparison with MCA

To illustrate the utility of the proposed co-clustering approach, let us consider real-world data sets for which the key characteristics are well known or from which conclusions may follow common sense, namely the Iris and Adult data sets (Lichman 2013). We start the experiments by comparing our methodology (Section 3.2) with MCA (Section 3.4) using the Iris data set for didactic reasons, then we apply our approach to the Adult data set to evaluate its scalability, while comparing our conclusions with those obtained using MCA followed by a clustering.

3.5.1 The Iris data set

Fisher's Iris data (Fisher 1936) is an example of easily exploitable data sets and for which key properties are well known. The data is available from the UCI Machine Learning Repository (Lichman 2013) and it consists of 150 instances, 750 observations, 4 numerical variables (*PetalLength*, *PetalWidth*, *SepalLength*, and *SepalWidth*) and 1 categorical variable (*Class*). The primary prediction task for this data set is to determine the class of a flower given the four numerical variables. However, we do not use this information in our co-clustering and to our approach, the data simply contains 5 variables of equal importance. As a consequence, the class information can be used to validate the obtained results.

The co-clustering results

After discretizing the Iris data using a granularity of $p = 5$ parts and applying the MODL co-clustering method (as explained in Section 3.2), we found that the optimal co-clustering consists of $G_u = 3$ clusters of instances and $G_p = 8$ clusters of variable parts as shown in Table 3.3 and Figure 3.1. Table 3.3 shows the table of counts containing the number of observations per co-cluster, along with the marginal counts per cluster. Figure 3.1 and Table 3.4 illustrate the co-clustering in terms of mutual information.

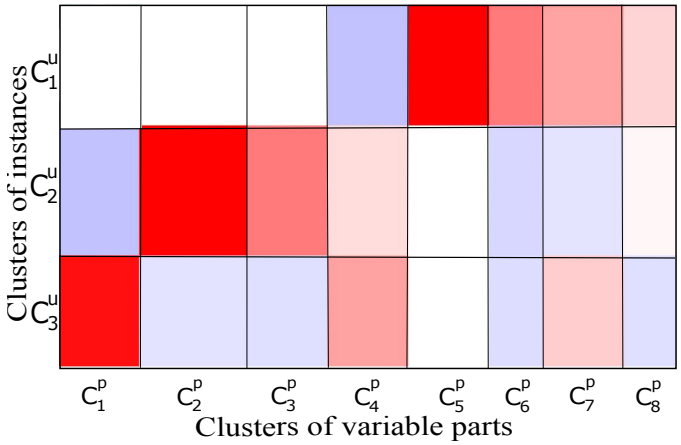


Figure 3.1 – *The resulting co-clusters as mutual information. The rows represent the instance clusters while the columns represent the variable part clusters. In each cell, the red color represents an over-representation of the instances compared to the case where the two dimensions are independent and the blue color represents an under-representation. White cells represent empty co-clusters (no association between the corresponding clusters).*

Composition of the clusters. The clusters of instances contain 50, 54, and 46 instances. The clusters of variable parts illustrate the association between parts of the original variables which in the correspondence analysis vocabulary would be seen as association or co-existence of different categories coming from different variables. If we denote C_k^u the k^{th} cluster of instances

Cluster	C_1^p	C_2^p	C_3^p	C_4^p	C_5^p	C_6^p	C_7^p	C_8^p	$N_{k.}$
C_1^u	0	0	0	11	121	49	48	21	250
C_2^u	8	131	64	46	0	4	1	16	270
C_3^u	109	1	21	56	0	3	34	6	230
$N_{.l}$	117	132	85	113	121	56	83	43	$N = 750$

Table 3.3 – *The contingency-table representation of Iris. C_k^u denotes the k^{th} cluster of instances and C_l^p denotes the l^{th} cluster of variable parts.*

Cluster	C_1^p	C_2^p	C_3^p	C_4^p	C_5^p	C_6^p	C_7^p	C_8^p	*marginal*
C_1^u	0	0	0	-0.018	0.177	0.063	0.035	0.011	0.268
C_2^u	-0.018	0.177	0.063	0.008	0	-0.009	-0.004	0.001	0.218
C_3^u	0.161	-0.005	-0.006	0.036	0	-0.007	0.013	-0.006	0.186
marginal	0.143	0.172	0.057	0.026	0.177	0.047	0.044	0.006	0.672

Table 3.4 – *Table of mutual information of the Iris data co-clustering.*

and C_l^p, the l^{th} cluster of variable parts, the compositions of the clusters of instances and of variable parts are given by Tables 3.5 and 3.6.

Cluster	$\lvert C_i^u \rvert$
C_1^u	50
C_2^u	54
C_3^u	46

Table 3.5 – *Composition of the instance clusters.*

Exploratory analysis of the results

The obtained co-clustering is easily exploitable because the data set is small. Therefore, no model coarsening is required and we will analyze the optimal co-clustering as it is.

Interpretation of the clusters. By ranking the clusters of variable parts by their contributions to each cluster of instances, we conclude the following associations: $(C_1^u, C_5^p), (C_2^u, C_2^p)$, and (C_3^u, C_1^p). From these associations, one can conclude that the data set contains three clusters of instances and their characteristics are as follows:

- a cluster C_1^u of 50 flowers that are characterized by C_5^p (i.e. the variable parts $Class\{setosa\}$, $PetalLength]-\infty; 1.55]$ and $PetalWidth]-\infty; 0.25]$),
- a cluster C_2^u of 54 flowers characterized by C_2^p (i.e. the variable parts $Class\{virginica\}$, $PetalLength]5.35; +\infty[$, $PetalWidth]1.95; +\infty[$ and $PetalWidth]1.55; 1.95]$),

Cluster	C_1^p	C_2^p	C_3^p	C_4^p
	$Class\{versicolor\}$	$Class\{virginica\}$	$PetalLength]4.65; 5.35]$	$SepalWidth]2.75; 3.05]$
	$PetalWidth]1.15; 1.55]$	$PetalLength]5.35; +\infty[$	$SepalLength]6.55; +\infty[$	$SepalWidth]-\infty; 2.75]$
	$PetalLength]3.95; 4.65]$	$PetalWidth]1.95; +\infty[$	$SepalLength]6.15; 6.55]$	$SepalLength]5.65; 6.15]$
		$PetalWidth]1.55; 1.95]$		
Cluster	C_5^p	C_6^p	C_7^p	C_8^p
	$Class\{setosa\}$	$SepalLength]-\infty; 5.05]$	$SepalLength]5.05; 5.65]$	$SepalWidth]3.15; 3.45]$
	$PetalLength]-\infty; 1.55]$	$SepalWidth]3.45; +\infty[$	$PetalWidth]0.25; 1.15]$	$SepalWidth]3.05; 3.15]$
	$PetalWidth]-\infty; 0.25]$		$PetalLength]1.55; 3.95]$	

Table 3.6 – *Composition of the variable part clusters.*

- a cluster C_3^u of 46 flowers characterized by C_1^p (i.e. the variable parts $Class\{versicolor\}$, $PetalLength]3.95; 4.65]$ and $PetalWidth]1.15; 1.55]$).

These three instance clusters are easily understandable as they represent the *small*, *large* and *medium* flowers respectively. These clusters are mainly explained by three clusters of variable parts containing the variables *Class*, *PetalLength* and *PetalWidth*. In fact it is well known that, in the Iris data set, the three classes are well separated by the Petal variables. This is reflected here by the grouping of the variables as well as by the instance clusters.

Non-informative clusters of variable parts. By looking at the contribution of the clusters of variable parts to mutual information, one can distinguish two non informative clusters (the fourth and eighth columns C_4^p and C_8^p of Figure 3.1 and Table 3.4), which are based essentially on the variable *SepalWidth*:

- the fourth column C_4^p contains the parts: $SepalWidth]-\infty; 2.75]$, $SepalWidth]2.75; 3.05]$, and $SepalLength]5.65; 6.15]$,
- the eighth C_8^p column contains the parts: $SepalWidth]3.05; 3.15]$ and $SepalWidth]3.15; 3.45]$.

The small values of *SepalWidth* (C_4^p) are slightly over-represented in the clusters of instances associated to the classes *versicolor* and *virginica* while the intermediate values (C_8^p) are slightly over-represented in the cluster of instances associated to *setosa*.

Notice that, as expected, the methodology enables us to group values of different nature in the same cluster. However, it does not leverage the origins of the variable parts nor their inter-relations. For example, the variable parts $PetalWidth]1.55; 1.95]$ and $PetalWidth]1.95; +\infty[$ belong to the same cluster but, since they both come from the variable *PetalWidth* and are contiguous, their presence in the same cluster is equivalent to having the variable part $PetalWidth]1.55; +\infty[$ that groups them both.

MCA analysis

For comparison, we perform a multiple correspondence analysis approach to the discretized data resulting from the pre-processing step. The distribution

of eigenvalues (Figure 3.2) indicates that the first two principal axes do capture enough information with a cumulative variance of 38.30%. Therefore, we can limit our analysis to the first factorial plan.

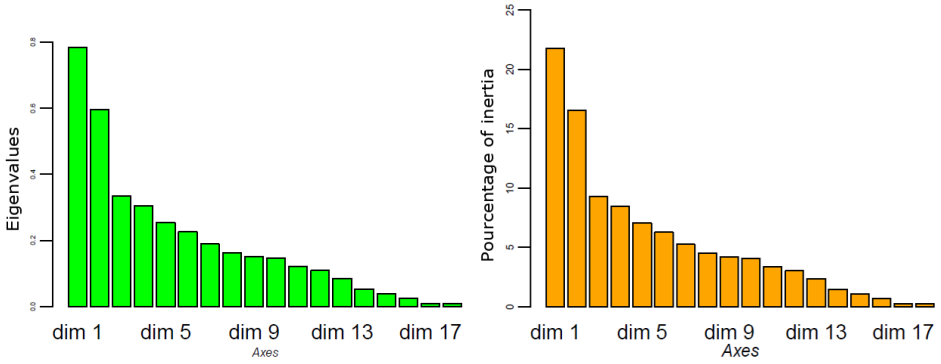
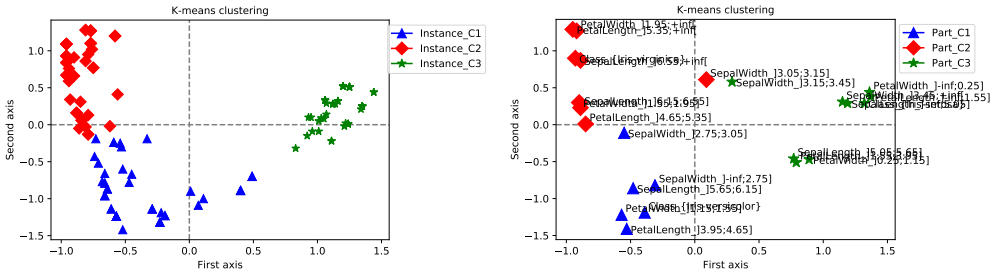


Figure 3.2 – *Histogram of eigenvalues (on the left) and the percentage of variance captured by the axes in the MCA analysis of Iris (on the right).*



(a) *Projection of the instances* (b) *Projection of the variable parts*

Figure 3.3 – *K-means clustering of the projection of the set of instances and variable parts.*

To exploit the results, we perform a k-means clustering on the projections of the instances and the variable parts on the factor space formed by the first two axes. Figure 3.3 shows the projection of the instances and the variable parts by their k-means clusters with $k = 3$, which is the number of clusters discovered by our approach.

A comparison between the k-means clustering and the clustering resulting from our co-clustering shows a nearly perfect correspondence (Table 3.7).

Analysis of the results. The k-means clustering enables us to make the following conclusions:

1. The first cluster of instances (in the top left corners of Figure 3.3), containing 51 instances, is associated to the variable parts:

   - $SepalWidth]3.05; 3.15]$
   - $SepalLength]6.15; 6.55]$ and $SepalLength]6.55; +\infty[$

Co-clustering *vs* K-means

Confusion	C_1	C_2	C_3
C_1^u	0	0	50
C_2^u	5	49	0
C_3^u	46	0	0

Table 3.7 – *Confusion table between our clustering and the k-means clustering. C_i stands for the i^{th} k-means cluster.*

- $PetalLength]4.65; 5.35]$ and $PetalLength]5.35; +\infty[$
- $PetalWidth]1.55; 1.95]$ and $PetalWidth]1.95; +\infty[$
- $Class\{virginica\}$.

Thus associating *virginica* with high values of *PetalLength* (greater than 4.65), high values of *PetalWidth* (greater than 1.55) and high values of *SepalLength* (greater than 6.15).

2. The second cluster (on the right of Figure 3.3) associates 49 instances in the cluster to the variable parts:

   - $SepalLength] - \infty; 5.05]$, $SepalLength]5.05; 5.65]$,
   - $SepalWidth]3.15; 3.45]$, $SepalWidth]3.45; +\infty[$,
   - $PetalLength]1.55; 3.95]$,
   - $PetalWidth] - \infty; 0.25]$, $PetalWidth]0.25; 1.15]$,
   - $Class\{setosa\}$.

   Thus, strongly associating *setosa* with low values of *PetalLength* (less than 3.95), low values of *PetalWidth* (less than 1.15) and low values of *SepalLength* (less than 5.65).

3. The third cluster (in the bottom left corners of Figure 3.3) associates the 50 instances in the cluster to:

   - $SepalLength]5.65; 6.15]$, $SepalLength]6.15; 6.55]$,
   - $SepalWidth] - \infty; 2.75]$, $SepalWidth]2.75; 3.05]$,
   - $PetalLength]3.95; 4.65]$,
   - $PetalWidth]1.15; 1.55]$,
   - $Class\{versicolor\}$.

   Thus, associating *versicolor* with intermediate values of *PetalLength*, *PetalWidth* and *SepalLength*, and with low values of *SepalWidth*.

4. The projection of instances (on the left of figure 3.3) shows a mixture between *virginica* and *versicolor*. These results are identical to those found using the co-clustering analysis.

5. The variable parts issued from *SepalWidth* are weakly correlated with the others and contribute less to the first factorial plan: the small values (less than 3.05) are associated with the mixture zone between *virginica* and *versicolor*, the intermediate values (between 3.05 and 3.45) have their projections in between *virginica* and *setosa* (they are therefore present in both clusters). These results are also in agreement with the results deduced from the co-clustering (see the above interpretation of the clusters C_4^u and C_8^u).

In summary, on this didactic example where the results of MCA are easily interpretable, a good agreement emerges between a k-means clustering on the MCA projections and the proposed co-clustering approach.

3.5.2 The Adult data set

The Adult data set (Lichman (2013)) is composed of $I = 48.842$ instances represented by $K_n = 6$ numerical and $K_c = 9$ categorical ones. The variables are income related variables such as age, working class, education, sex, hours worked, and salary. The prediction task for this data set is to determine whether a person makes *more* or *less* than $50K$ a year (the variable *class*). However, we do not use this information in our co-clustering and to our model, the data simply contains $K = 15$ variables of equal importance. As a consequence, the class information can be used to validate the obtained results.

The co-clustering results

For this data set we choose $p = 10$. When the Adult data is discretized, and the transformation into two variables is performed as presented previously, we obtain a data set of $N \approx 750,000$ rows and two columns: the *IdInstance* variable containing around $I \approx 50,000$ values (corresponding to the initial instances) and the *IdVarPart* variable containing $K \times p \approx 150$ values (corresponding to the variable parts). The result of the co-clustering is shown in Figure 3.4 in terms of mutual information.

Exploratory analysis of the results

The obtained result is very detailed, with 34 clusters of instances and 62 clusters of variable parts. In an exploratory analysis context, this level of detail hinders the interpretability. Thus, we start by simplifying the co-clustering structure by iteratively merging the rows and columns of the finest co-clustering until reaching a reasonable percentage of the initial amount of information (see Section 3.3).

Model coarsening. For this analysis, we choose to extract conclusions at two levels of granularity. Namely, from the simplified co-clustering given in Figure 3.5, which contains 10×14 co-clusters and captures 70% of the initial

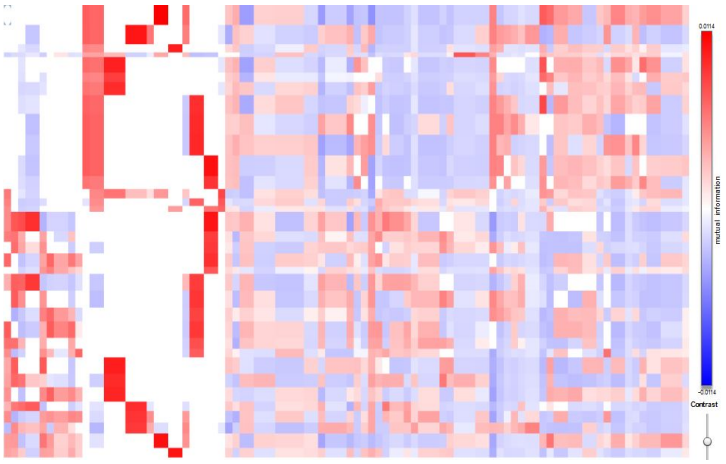


Figure 3.4 – *Co-clustering of the Adult data set. Each square represents a co-cluster. This MODL optimal co-clustering contains* 34×62 *co-clusters.*

information in the data, and from a simplified co-clustering that contains two clusters of instances as in Figure 3.6. In the former the merging have been performed on both dimensions where in the latter we keep 14 clusters of variable parts and merge on the dimension of instances.

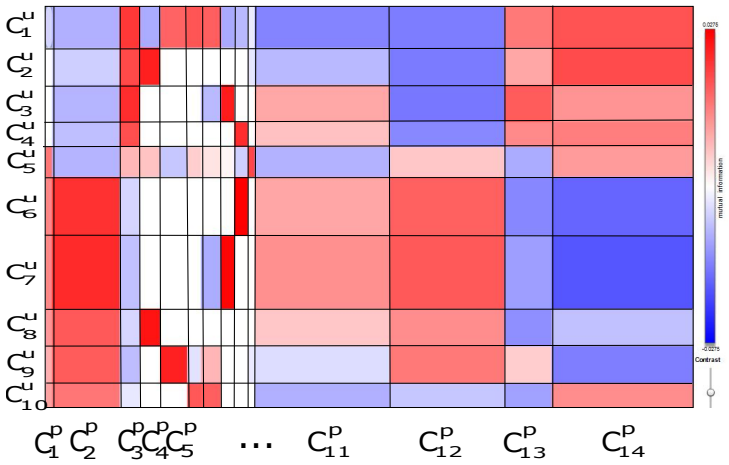


Figure 3.5 – *A simplified co-clustering of the Adult data set, with 70% of information. The rows represent clusters of instances while the columns represent clusters of variable parts.*

The composition of the clusters of variable parts is shown in Table 3.10. From the co-clustering results we make the following conclusions.

1. The first level of retrieved patterns appears clearly when we consider dividing the clusters of instances into two parts, visible on the top half and the bottom half of the co-clustering cells presented in Figure 3.6. The instance clusters in the top half are mainly men with a good salary, with an over-representation of the variable part clusters containing: *sex{Male}*, *relationship{Husband}*, *relationship{Married...}*, *class-{More}*, *age*$]45.5; 51.5]$, *age*$]51.5; 58.5]$, *hours_per_week*$]48.5; 55.5]$,

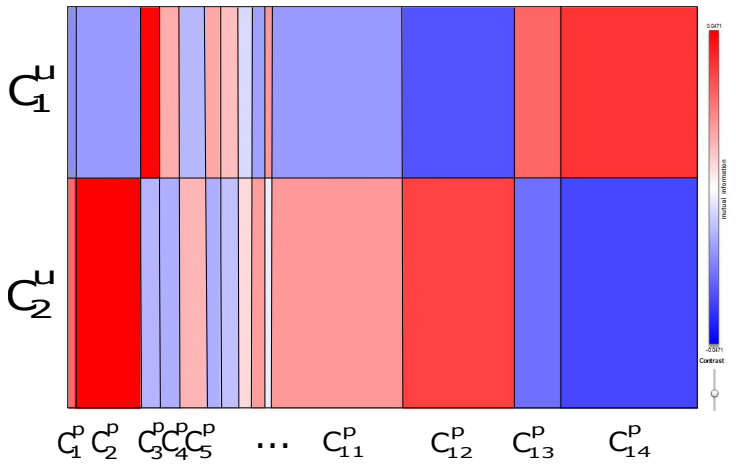


Figure 3.6 – *A simplified co-clustering of the Adult data set, with* 2×14 *co-clusters.*

hours_per_week$]55.5; +\infty[$. The instance clusters in the bottom half are mainly for women or rather poor unmarried men, with an over-representation of the variable part clusters containing: *class-{Less}*, *sex{Female}*, *maritalStatus{Never-married}*, *maritalStatus-{Divorced}*, *relationship{Own-child}*, *relationship{Not-in-family}*, *relationship{Unmarried}*.

2. From the optimal co-clustering (Figure 3.4), the most informative clusters of instances is on the first row and it can be interpreted by the over-represented variable part clusters in the same row:

   - *relationship{Husband}*, *relationship{Married...}*,
   - *educationNum*$]13.5; +\infty[$, *education{Masters}*,
   - *education{Prof-school}*,
   - *sex{Male}*,
   - *class{more}*,
   - *occupation{Prof-specialty}*,
   - *age*$]45.5; 51.5]$, *age*$]51.5; 58.5]$,
   - *hours_per_week*$]48.5; 55.5]$, *hours_per_week*$]55.5; +\infty[$.

   It is therefore a cluster of around 2000 instances, with mainly married men with rather long studies, working in the field of education, at the end of their careers, working extra-time with good salary.

3. From the simplified co-clustering (Figure 3.5), the most contrasted clusters of variable parts, hence the most informative, are those presented by the columns C_4^p to C_9^p. These contain only variable parts issued from the variables *education* and *educationNum* which are the most correlated variables in this data set. The compositions of these clusters are as follows.

- *educationNum*$]11.5; 13.5]$, *education{Assoc-acdm}*, *education{Bachelors}* (the 4^{th} column),
- *educationNum*$]-\infty; 7.5]$, *education{10th}*, *education{11th}*, *education{7th-8th}*(the 5^{th} column),
- *educationNum*$]13.5; +\infty[$, *education{Masters}*(the 6^{th} column),
- *educationNum*$]10.5; 11.5]$, *education{Assoc-voc}*, *education{Prof-school}* (the 7^{th} column),
- *educationNum*$]7.5; 9.5]$, *education{HS-grad}*(the 8^{th} column),
- *educationNum*$]9.5; 10.5]$, *education{Some-college}*(the 9^{th} column).

This illustrates that the variables *education* (categorical) and *educationNum* (numerical) are very correlated as their variable part clusters seem particularly consistent.

MCA analysis

Figure 3.7 shows the distribution of the variability captured by the axes along with the cumulative level of information.

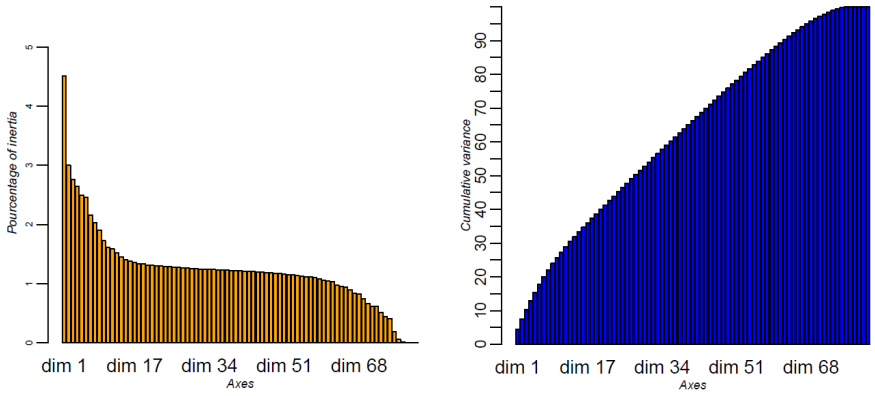


Figure 3.7 – *Barplots of the variability (on the left) and the cumulative information captured by the axes (on the right) in the MCA analysis of Adult.*

On the contrary to the smaller Iris data set, the distribution of the variance (Figure 3.7) indicates that the first two principal axes capture a cumulative variance of only 7.5%. Figure 3.8 shows the projections of the instances and variable parts on the first factorial plan where in the left side figure, the black circles are the instances that gain less than 50K and the red triangles are the instances that gain more than 50K. Without the prior knowledge about the class of each instance, which is the case in exploratory analysis, the projection of instances appears as a single dense cluster.

The projection on the first factorial plan does not allow to distinguish any clusters, which is not surprising given the low level of variability captured by this plan. However, in order to capture 20%, 25% or 30% of the variance, one needs to choose 7, 10 or 13 axes, respectively. Choosing a high number

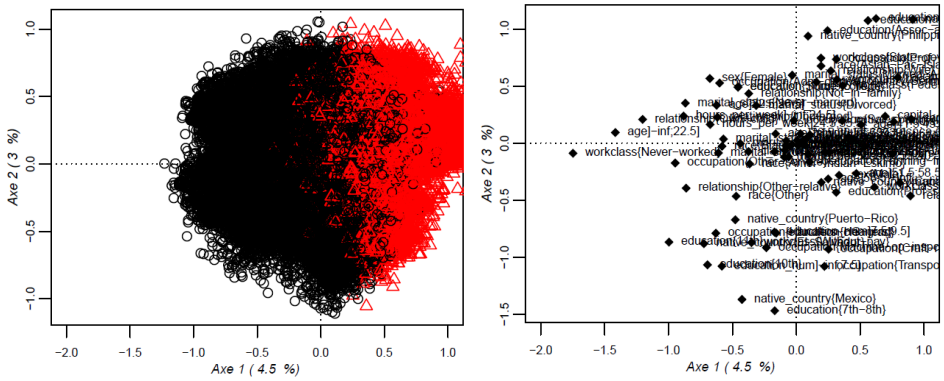


Figure 3.8 – *Projection of the set of instances and variable parts, of the Adult data set, on the first factorial plan.*

of axes, say 13, means that some post analysis of the projections is required not just for partitioning but also for visualization.

K-means of the MCA projections. In order to extract potentially meaningful clusters from the MCA results, we perform a k-means on the projections of the instances and the variable parts on the factor space formed by the first 13 axes. Figure 3.9 shows the projection of the k-means centers with $k = 10$ (on the left) and $k = 100$ (on the right to illustrate how complex the data is).

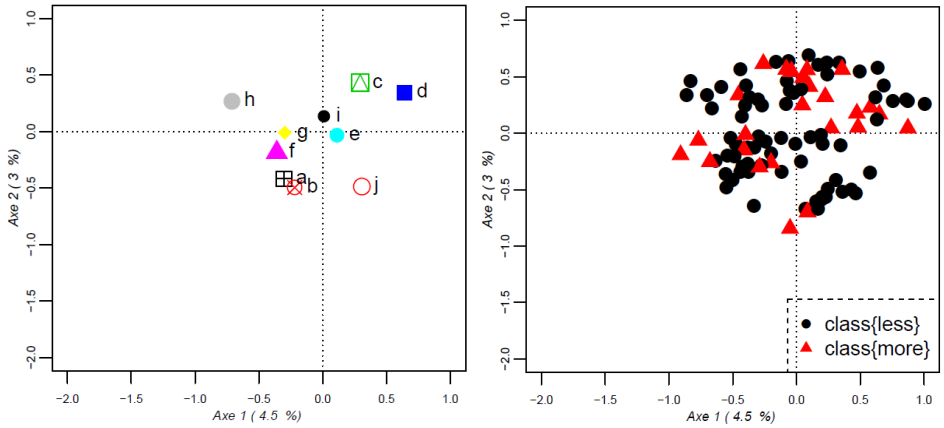


Figure 3.9 – *Projection of the k-means centers with k=10 and k=100 clusters, on the first factorial plan.*

The k-means clustering of the projections with $k = 2$ gives two clusters containing 26178 instances associated to 50 variable parts, and 22664 instances associated with 46 variable parts, respectively. The first cluster of instances associates the variable part *class{more}* with being married, white, a men, having more than 10.5 years of education, being more than 30.5 years old, working more than 40.5 hours per week, or originating from Canada, Cuba, India or Philippines. The second cluster of instances associates the variable part *class{less}* with being young $(age] - \infty; 30.5])$, having

less than 10.5 years of education, being never married, divorced or widowed, being Amer-Indian-Eskimo, black or another non white race (*race{Other}*), working for less than 40.5 hours per week, being a women or originated from countries like El-Salvador, England, Germany, Mexico, Puerto-Rico, and United-States. These clusters are consistent with the two main clusters found by the co-clustering, particularly in combining being a men, married, middle aged and working extra hours with earning more than 50K and associating being a women, never married, divorced, or having a child with earning less than 50k.

Table 3.8 shows a summary of the k-means clustering with $k = 10$ indicating the contribution of each cluster to the intra-cluster variance. To avoid confusion with the clusters resulting from co-clustering, we name the k-means clusters using letters: $\{a, b, c, d, e, f, g, h, i, j\}$.

cluster	a	b	c	d	e	f	g	h	i	j
size	4297	1572	9325	4033	2061	7686	1581	4075	7163	7049
withinss	4484.7	1849.6	8185.9	3919.8	1738.8	5156.8	1720.7	2490.5	5447.2	3701.6
withinss%	11.58	4.77	21.15	10.12	4.49	13.32	4.44	6.43	14.07	9.56

Table 3.8 – *Summary of the clusters of instances using k-means.*

Table 3.9 shows the confusion matrix between the clusters issued from the co-clustering method and the clusters issued from the k-means of projections.

The problem of comparing the two clusterings can be seen as a maximum weight matching problem in a weighted bipartite graph, also known as the assignment problem. It consists of finding the one-to-one matching between the nodes that provides a maximum total weight. This assignment problem can be solved using the Hungarian method Kuhn and Yaw (1955). Applied to the matrix of mutual information (derived from Table 3.9), the Hungarian algorithm results in the following cluster associations: (C_1^u, d), (C_2^u, g), (C_3^u, j), (C_4^u, i), (C_5^u, b), (C_6^u, h), (C_7^u, f), (C_8^u, c), (C_9^u, a), (C_{10}^u, e) as highlighted in Table 3.9. These same associations are also obtained when applying the algorithm to the χ^2 table. This one-to-one matching carries 76.3% of the total mutual information. The highest contributions to the conserved mutual information associate the k-means cluster a with the co-

cluster	a	b	c	d	e	f	g	h	i	j
C_1^u	1679	444	141	**2289**	886	12	7	0	48	111
C_2^u	0	20	4096	0	0	0	**0**	0	0	0
C_3^u	0	96	0	0	0	18	5	0	0	**6377**
C_4^u	0	31	0	0	0	0	0	13	**3588**	0
C_5^u	114	**88**	576	247	129	331	54	28	455	434
C_6^u	0	59	0	0	0	0	252	**3314**	3072	0
C_7^u	1	183	0	1	0	**7318**	776	609	0	127
C_8^u	0	27	**4512**	0	0	3	150	93	0	0
C_9^u	**2503**	617	0	0	0	2	299	16	0	0
C_{10}^u	0	7	0	1496	**1046**	2	38	2	0	0

Table 3.9 – *The confusion matrix between the co-clustering and k-means partitions.*

clustering cluster C_9^u, the k-means cluster c with the co-clustering cluster C_8^u, the k-means cluster f with the co-clustering cluster C_7^u, the k-means cluster h with the co-clustering cluster C_6^u, the k-means cluster i with the co-clustering cluster C_4^u, the k-means cluster j with the co-clustering cluster C_3^u. In terms of variable parts, these clusters are as follows:

- the cluster a contains individuals who never-worked or work as handlers-cleaners, have less than 7.5 years of education, or have a level of education from the 7th to the 11th grade,
- the cluster c contains instances characterized by: *workclass{Self-emp-inc}*, *education{Assoc-acdm}*, *education{Bachelors}*, *education_num*$]11.5; 13.5]$, *occupation{Exec-managerial}*, *occupation{Sales}*, *race{Asian-Pac-Islander}*, *capital_loss*$]77.5; +\infty[$, *hours_per_week*$]40.5; 48.5]$, *hours_per_week*$]48.5; 55.5]$, *native-country{Germany}*, *native-country{Philippines}*,
- the cluster f contains instances characterized by: earning less than 50K (*class{less}*), being relatively young (age$]26.5; 33.5]$), having relatively low level of education (*education{HS-grad}* and *education_num*$]7.5; 9.5]$), being unmarried, divorced or separated, being an Amer-Indian-Eskimo, Black or Female,
- the cluster h contains instances that work less than 35.5 hours per week, are under 26.5 years old, never married and have a child,
- the cluster i contains middle-aged individuals (between 41.5 and 45.5 years old), with moderate education (9.5 to 10.5 years of education) and working in farming or fishing,
- the cluster j contains instances characterized by the variable parts: *age*$]33.5; 37.5]$, *age*$]37.5; 41.5]$, *age*$]45.5; 51.5]$, *age*$]51.5; 58.5]$, *workclass {Self-emp-not-inc}*, *fnlwgt*$]65739; 178144.5]$, *hours_per_week*$]55.5; +\infty[$, *relationship{Husband}*, *marital_status {Married-AF-spouse}*, *marital_status {Married-civ-spouse}*, *sex{Male}*, *race{White}*, *occupation{Craft-repair}*, *occupation{Transport-moving}*.

To summarize, the clusters obtained using a k-means on the projections of the MCA, are somewhat consistent with those obtained using the co-clustering.

Discussion. An important contribution of our methodology, compared to MCA, is its ease of application and the direct interpretability of its results. When MCA is applied to a data set of significant size, such as Adult, the projections of instances and variables on the first factorial plan (and even on the second plan) do not enable us to distinguish any particularly dense clusters. Therefore, it is necessary to choose a high number of axes in order to capture enough information. On the Adult data set, we found that 13

axes explain only 30% of the information. Choosing this high number of axes means that some post analysis of the projections (such as k-means) is necessary to visualize the clusters. Using this approach, the results obtained using k-means, although only explaining 30% of the information, are consistent with those obtained using the co-clustering using our two-step methodology. However, with our methodology, the hierarchy of clusters enables us to choose the desired level of detail and the percentage of information, then one can distinguish, and eventually explain, the most informative clusters, by their contribution to the total information (Section 3.3).

3.6 Conclusion

This chapter have proposed a methodology for using co-clustering in exploratory analysis of mixed-type data. The methodology consists in homogenizing the variables, then applying a standard value based co-clustering approach to the transformed homogeneous data. For co-clustering, we have used the MODL approach (Boullé (2011)) which have the advantage of automatically inferring the numbers of clusters.

We have shown that the exploratory analysis of the co-clustering reveals a good agreement between MCA and co-clustering, despite the differences between the models and the methodologies. We have also shown that the proposed approach can be applied to significantly large and complex data sets.

However, this methodology is limited by the need for the analyst to choose the number of parts per variable, used for data discretization, and the discretization method. Furthermore, the co-clustering method does not follow the origins of the parts which would be useful in order to take into account the intrinsic correlation structure that exists between the parts originating from the same variable, forming a partition. In the next chapter, we will handle these limitations by defining a co-clustering model that integrates the granularity parameter and tracks the variable parts that form a partition of the same variable.

C_1^p: sex$\{Female\}$	C_3^p: $marital\_status\{Married - civ - spouse\}$ $relationship\{Husband\}$ $marital\_status\{Married - AF - spouse\}$
C_2^p: $relationship\{Not - in - family\}$ $relationship\{Own - child\}$ $relationship\{Unmarried\}$ $age]-\infty; 22.5]$ $marital\_status\{Never - married\}$ $marital\_status\{Divorced\}$ $marital\_status\{Separated\}$ $marital\_status\{Widowed\}$ $marital\_status\{Married - spouse - absent\}$ $workclass\{Never - worked\}$	C_4^p: $education\_num]11.5; 13.5]$ $education\{Bachelors\}$ $education\{Assoc - acdm\}$
	C_5^p: $education\_num]-\infty; 7.5]$ $education\{11th\}$ $education\{10th\}$ $education\{7th - 8th\}$
C_6^p: $education\_num]13.5; +\infty[$ $education\{Masters\}$	C_{11}^p: $capital\_loss]-\infty; 77.5]$, $capital\_gain]-\infty; 57]$ $native\_country\{United - States\}$ $race\{White\}$, $workclass\{Private\}$ $hours\_per\_week]35.5; 40.5]$ $fnlwgt]130708.5; 157936.5]$ $fnlwgt]178144.5; 196318]$ $fnlwgt]220158; 260259.5]$ $fnlwgt]260259.5; 328492]$ $fnlwgt]-\infty; 65739]$ $fnlwgt]196318; 220158]$ $fnlwgt]106068.5; 130708.5]$ $fnlwgt]157936.5; 178144.5]$ $fnlwgt]65739; 106068.5]$, $fnlwgt]328492; +\infty[$ $occupation\{Sales\}$, $occupation\{Farming - fishing\}$ $native\_country\{Germany\}$, $native\_country\{Cuba\}$
C_7^p: $education\{Prof - school\}$ $education\_num]10.5; 11.5]$ $education\{Assoc - voc\}$	
C_8^p: $education\_num]7.5; 9.5]$ $education\{HS - grad\}$	
C_9^p: $education\_num]9.5; 10.5]$ $education\{Some - college\}$	
C_{10}^p: $relationship\{Wife\}$	
C_{12}^p: $class\{less\}$ $occupation\{Adm - clerical\}$ $hours\_per\_week]24.5; 35.5]$ $age]26.5; 30.5]$ $hours\_per\_week]-\infty; 24.5]$ $occupation\{Other - service\}$ $age]22.5; 26.5]$ $race\{Black\}$	C_{12}^p (continued): $native\_country\{England\}$ $race\{Asian - Pac - Islander\}$ $relationship\{Other - relative\}$ $race\{Amer - Indian - Eskimo\}$ $race\{Other\}$ $native\_country\{Philippines\}$ $native\_country\{Puerto - Rico\}$, $native\_country\{India\}$ $workclass\{Without - pay\}$
C_{13}^p: $sex\{Male\}$ $occupation\{Craft - repair\}$ $occupation\{Machine - op - inspct\}$ $occupation\{Transport - moving\}$ $occupation\{Handlers - cleaners\}$ $native\_country\{Mexico\}$ $native\_country\{El - Salvador\}$	C_{14}^p: $class\{more\}$ $occupation\{Prof - specialty\}$ $occupation\{Exec - managerial\}$ $age]45.5; 51.5]$,$age]33.5; 37.5]$, $age]37.5; 41.5]$ $age]58.5; +\infty[$, $age]41.5; 45.5]$, $age]51.5; 58.5]$ $age]30.5; 33.5]$ $hours\_per\_week]40.5; 48.5]$ $hours\_per\_week]48.5; 55.5]$ $hours\_per\_week]55.5; +\infty[$ $capital\_gain]57; +\infty[$, $capital\_loss]77.5; +\infty[$ $workclass\{Self - emp - not - inc\}$, $workclass\{Local - gov\}$ $workclass\{State - gov\}$, $workclass\{Self - emp - inc\}$ $workclass\{Federal - gov\}$ $native\_country\{Canada\}$

Table 3.10 – *Composition of the clusters of variable parts in the simplified Adult co-clustering.*

Chapter 4

A new co-clustering model for mixed type data

In the previous chapter, we have proposed a co-clustering methodology for mixed type data. The approach consists of homogenizing the data to create categorical variable parts then using the MODL approach for co-clustering. This approach requires specifying the number of parts per variable and uses equal frequency dicretization. In this chapter, we propose a general-purpose, integrated generative model, for co-clustering mixed data at the *instances* $\times$ *mixed variable parts* level. The proposed model automatically infers the number of parts per variable, creates an optimized partitioning of each variable, and performs a co-clustering, by optimizing a MAP based criterion.

This chapter is organized as follows. Section 4.1 introduces the motivation of this model, and the underlying assumptions. In particular, we introduce the notion of *observation*, which is the value of a measured feature for a given object and a given measurement, as the statistical unit. Section 4.2 details the basic components of the model including the set of parameters, the supposed co-clustering structure, the data generation mechanism, and the proposed model selection criterion. The optimization of the proposed criterion allows to simultaneously estimate the model parameters, the cluster memberships and the number of clusters on each dimension. For optimization, a greedy minimization algorithm is used. We conclude in Section 4.3.

4.1 Introduction

This chapter introduces a MAP based approach to perform co-clustering of mixed data. The goal of the proposed approach is to perform a clustering of the values in a data matrix through simultaneous clustering of the objects and variables. The approach we proposed in the previous chapter had the same goal but required a pre-processing step to discretize the data. The pre-processing results in a coarse approximation of the distribution of each variable individually. However, this partitioning of variables is not optimized. Hence the efficacy of the approach and its level of accuracy rely

on this dicretization. In this chapter, we propose a non parametric model that takes, as input, the data presented as values indexed by their relative instances and variables and outputs both an optimal partitioning of the variables, that takes into account their interdependence, and an optimal co-clustering of the values. As a result of this optimized partitioning, the approach performs a clustering of the objects while extracting the interdependence between the variables with respect to the clusters of objects, through optimized partitioning.

4.1.1 Data representation

Our goal is to co-cluster mixed data. Therefore, we rely on a specific data representation and on specific data considerations that enable us to handle numeric and categorical data simultaneously.

Instances and variables. We consider an instance to be a representation of an object in the physical world. Assume a set of I instances, denoted $\mathcal{U} = u_1, \ldots, u_I$, described by K variables denoted $\mathcal{X} = \{X_1, \ldots, X_K\}$. Among these variables, suppose K_n are quantitative (i.e. take values in $\mathbb{R}$) and form the set $\mathcal{X}_n$. The remaining K_c variables are qualitative (i.e. take values in finite sets) and form the set $\mathcal{X}_c$. For any variable X_k, $\mathcal{V}_k$ denotes the set of possible values for this variable, its domain.

In the remaining of this chapter, we use the term *instance* to denote a real world *object*. We also use the terms *categorical* and *qualitative*, *numerical* and *quantitative*, interchangeably, to describe the variables.

Observations. The proposed model performs a co-clustering of the set of observations. An observation occurs when an instance takes a value for a variable. In general, not all instances have to be observed at the same time for all variables. Also, a couple instance-variable can be observed multiple times, resulting in a series of values for the same instance. Therefore, we represent the data by a set of N observations $\mathcal{O} = \{o_1, \ldots, o_N\}$. Each observation is a triple $o_l = (u, k, v)$ where u is an object, k is a variable index and v is an element of $\mathcal{V}_k$. The observation o_l means that the value of variable X_k is v for object u, for a given measurement. This arguably complex representation has two initial advantages over a classical tabular representation, namely counting for missing values and set valued variables. Table 4.1 shows an example of data for which the model is proposed, containing $K_n = 3$ numerical variables, $K_c = 2$ categorical variables, and $N = 26$ observations.

Ranks of the numerical values. In order to avoid the limitations of a parametric approach for modeling the distributions of the quantitative variables, we replace their values by their rank in the data set, variable per variable. Thus, when $X_k \in \mathcal{X}_n$, then $\mathcal{V}_k \in \{1, \ldots, N\}$ (see Section 4.2.5 for

	X_1	X_2	X_3	X_4	X_5
$u_1 \rightarrow$	0	-1	.	$\{b, a\}$	A
$u_2 \rightarrow$	3	$\{0.2, 1, 0\}$	0	b	B
$u_3 \rightarrow$	2	.	5	$\{a, c\}$	A
$u_4 \rightarrow$	.	2	22	c	C
$u_5 \rightarrow$	5	3	24	d	C

Table 4.1 – *Data example.*

an example). When providing results of the method, we would revert the ranks to the original values which are easier to interpret.

4.2 The co-clustering model

The co-clustering model is composed of a mapping of the variables to ranges of values (called variable parts), a mapping of the instances to clusters of instances, a mapping of the variable parts to clusters of variable parts, and a mapping of the observations to co-clusters.

4.2.1 Variable parts

In order to be able to group observations of different variable types in the same cluster, we require partitions of the variable domains. That is, a transformation of the variable values (the $\mathcal{V}_k$ sets) into dense categories (parts). Then, we would perform a simultaneous grouping of the instances and the new categories. More precisely, we suppose the existence of K partitions of the variable domains $\{P_1, \ldots, P_K\}$ that, on the uni-variate level, approximate the densities of the variables, and globally allow for a better detection of the associations between the variables with respect to a clustering of the instances. By association, we mean that, ideally, the variables should be partitioned based on their mutual interdependence relationships.

A variable partition is defined by the number of parts and the actual partitioning of the observed values. A partition of size J_k is a grouping of the observed values into J_k groups if the variable is categorical ($X_k \in \mathcal{X}_c$) and a partition of the ranks of the observed values into J_k contiguous intervals if the variable is numerical ($X_k \in \mathcal{X}_n$). In the following, each class of those partitions will be called a *variable part* and it is naturally associated to an indicator variable $X_{k,c}$. Indeed if $c \in P_k$ is a class of P_k (in other word, a variable part), $X_{k,c}$ is the variable with values in $\{0, 1\}$ which takes value 1 if and only if X_k takes a value in c. The indicator variables can be used as a way to keep track of an easy mapping between the *variable parts* and the *original variables*.

The variable partitions are a model parameter. They need to be estimated with respect to the *clusters of objects* and with respect to inter-variable associations, which is expressed by the *clusters of variable parts*.

4.2.2 Co-clusters

Now that all variables are represented by their approximating partitions, the main goal is to co-cluster the instances and variables. Clearly, this co-clustering acts as a second level of clustering built upon the variable domain partitions. As such, it consists in a partition $\mathbf{C}^u$ of the instances and in a partition $\mathbf{C}^p$ of the variable parts. The former is a simple partition of the set of instances $\mathcal{U} = \{u_1, \ldots, u_I\}$ while the latter is a partition of the set of partitions $\mathcal{P} = P_1 \cup \ldots \cup P_K$. The model can therefore be interpreted as a co-clustering model on transformed data, i.e. where the original variables would have been replaced by the indicator variables associated to variable parts.

The co-clustering is defined by the co-clustering structure and the actual co-clustering of the instances and variables. The co-clustering structure is defined by the number of clusters of instances and the number of clusters of variable parts. The act of co-clustering is defined by distributing the set of observations on the co-clustering structure. Clearly, an arbitrary distribution is of no use to the task of extracting knowledge from the data. Therefore, a well controlled distribution is needed, along with a measure for evaluating its fit to the data.

The co-clustering structure (the partitions $\mathbf{C}^u$ and $\mathbf{C}^p$) is a model parameter to be estimated.

4.2.3 The model parameters

The co-clustering model of mixed-type is based on: variable partitions, a partition $\mathbf{C}^u$ of the instances into clusters, and a partition $\mathbf{C}^p$ of the variable parts into clusters. The model is defined using a hierarchy of the parameters. At each stage of the hierarchy, the parameters are chosen with respect to the previous ones.

The model parameters. The co-clustering model of mixed-type data is built upon the following parameters:

1. ϕ the number of observations to generate,
2. μ the number of instances,
3. a number of parts J_k for each variable X_k,
4. for each qualitative variable $X_k \in \mathcal{X}_c$, a partition $P_k = \{P_{k,1}, \ldots, P_{k,J_k}\}$ of its values into the chosen number of parts J_k. We will show later that the partitions of the ranks of the numerical variables do not need to be defined as a model parameter.
5. a number of instance clusters, G_u,

6. a partition $\mathbf{C}^u$ of the set of instances $\mathcal{U}$ into the previously chosen number of instance clusters G_u. From this partition, let $m_{g_u}^{(u)}$ denote the number of instances in the g_u^{th} cluster of instances $C_{g_u}^u$.

7. a number of variable part clusters, G_p,

8. a partition $\mathbf{C}^p$ of $\mathcal{P} = P_1 \cup \ldots \cup P_K$ into the previously chosen number of variable part clusters G_p. From this partition, let $m_{g_p}^{(p)}$ denote the number of parts in the g_p^{th} cluster of variable parts $C_{g_p}^p$.

9. the distribution of the observations over the cells of the resulting instances×parts co-clusters. This distribution is represented by a matrix of counts $\mathbf{\Phi} = (\phi_{g_u,g_p})_{1\leq g_u\leq G_u,\, 1\leq g_p\leq G_p}$, giving the number of observations per co-cluster. More precisely for each co-cluster, formed by the cluster of instance $C_{g_u}^u$ and the cluster of variable parts $C_{g_p}^p$, the model will generate ϕ_{g_u,g_p} triples (u, k, v) with $u \in C_{g_u}^u$ and such that $P_{k,l} \in C_{g_p}^p$ and $v \in P_{k,l}$ for some part index l.

10. the vector $n^u = (n_1^u, \ldots, n_\mu^u)$ containing the number of observations per instance, giving the distribution of the observations of each cluster of instances over the instances in the cluster. We use $n_{g_u,i}^u$ to denote the number of observations associated to the instance u_i of the cluster $C_{g_u}^u$, and n_i^u when we refer to the i^{th} instance (within the set $\mathcal{U}$) without reference to a cluster.

11. the vector $n^p = (n_1^p, \ldots, n_J^p)$ containing the number of observations per variable part. We use $n_{k,l}^p$ to denote the number of observations associated to the l^{th} part of the variable X_k, and equivalently $n_{g_p,l}^p$ to denote the number of observations associated to the l^{th} part of the cluster of variable parts $C_{g_p}^p$. The vector n^p gives the distribution of the observations of each cluster of parts over the parts in the cluster.

12. for each qualitative variable $X_k \in \mathcal{X}_c$, the vector $n_k^v = (n_{k,1}^v, \ldots, n_{k,|\mathcal{V}_k|}^v)$. In this vector $n_{k,t}^v$ gives the number of observations of the form (u_t, k, v_t) that the model will generate, where v_t is the t-th value in $\mathcal{V}_k$. The vector n_k^v gives the distribution of the observations in each part of the variable X_k over the set of values that belong to the part. The set of vectors n_k^v form the set $n^v = (n_1^v, \ldots, n_{K_c}^v)$ which gives the distribution of the observations in each categorical variable part over the set of values that belong to the part.

Remark.

- Notice that, the number of instances per cluster of instances, the number of parts per cluster of variable parts, the number of observations per cluster of instances and per cluster of variable parts can all be deduced from the model parameters.

- In this model, clusters can be empty. This enables to decouple the number of clusters from the clustering itself. Notice also that contrarily to numerous classical models, the co-clustering structure is a parameter of the model, not a set of latent variables nor a user parameter.

4.2.4 Constraints over the parameters

The parameters described in the previous section must fulfill the following structural constraints:

$$\phi = \sum_{g_u=1}^{G_u} \sum_{g_p=1}^{G_p} \phi_{g_u,g_p}, \tag{4.1}$$

$$\forall g_u, \text{ the cluster } C^u_{g_u} \text{ satisfies} \quad \sum_{u_i \in C^u_{g_u}} n^u_{g_u,i} = \sum_{g_p=1}^{G_p} \phi_{g_u,g_p}, \tag{4.2}$$

$$\forall g_p, \text{ the cluster } C^p_{g_p} \text{ satisfies} \quad \sum_{l \in C^p_{g_p}} n^p_{g_p,l} = \sum_{g_u=1}^{G_u} \phi_{g_u,g_p}, \tag{4.3}$$

$$\forall\ X_k \in \mathcal{X}_c \qquad \sum_t n^v_{k,t} = \sum_{l=1}^{J_k} n^p_{k,l}, \tag{4.4}$$

$$\forall\ X_k \in \mathcal{X}_n, \forall l \qquad |P_{k,l}| = n^p_{k,l}. \tag{4.5}$$

The first equation matches the total number of observations to the counts per co-cluster. The second and third equations match marginal counts to joints. For instance, equation (4.2) says that, for each cluster of instances $C^u_{g_u}$, the total number of observations associated to the instances of this cluster (left hand part of the equation) must be equal to the total number of observations as specified over variable part clusters (right hand part).

Equation (4.4) matches per variable part count to the counts per value for qualitative variables. Equation (4.5) plays a similar role for quantitative variables but in a stricter way as the values of those variables are unique ranks. Notice that for these variables, the parts $P_{k,l}$ consist in intervals of values, and thus those constraints completely specify the partition.

In the rest of this chapter, Θ denotes a vector of parameters for the model that fulfills the above constraints:

$$\Theta = \{\phi, \mu, \{J_k\}, \{P_k\}_{X_k \in \mathcal{X}_c}, G_u, \mathbf{C}^u, G_p, \mathbf{C}^p, \Phi, n^u, n^p, n^v\}.$$

Individual components of Θ are referred to using the notations introduced in the previous section.

4.2.5 Illustrative data example

Let us consider a simple data set to illustrate our notations and the data representation by *observations*. Suppose we have $I = 5$ instances described by $K = 5$ variables in a standard representation as in Table 4.2.

Instance	X_1	X_2	X_3	X_4	X_5
u_1	0	-1	.	$\{b, a\}$	A
u_2	3	$\{0.2, 1, 0\}$	0	b	B
u_3	2	.	5	$\{a, c\}$	A
u_4	.	2	22	c	C
u_5	5	3	24	d	C

Table 4.2 – *A simple data set in its natural representation.*

Each line gives one of the 5 instances. A dot . stands for a missing value, while a set is used to denote several values for a given variable. For example, the instance u_1 has a missing value for the variable X_3 and has two values, b and a, for variable X_4.

Table 4.3 gives the observation based representation of the data set. We have $N = 26$ observations (taking into account missing values and set valued variables). This table illustrates several important aspects of the representation. The two values of variable X_4 for the instance u_1 are now represented by two observations (o_3 and o_4). Numerical variables X_1, X_2 and X_3 are represented via ranks rather than values. For example, as 0.2 is the third value for X_2, the corresponding observation is $o_7 = (u_2, 2, 3)$ rather than $o_7 = (u_2, 2, 0.2)$.

observation	instance	variable	value
o_1	u_1	1	1
o_2	u_1	2	1
o_3	u_1	4	b
o_4	u_1	4	a
o_5	u_1	5	A
o_6	u_2	1	3
o_7	u_2	2	3
$\vdots$			$\vdots$
o_{14}	u_3	3	2
$\vdots$			$\vdots$
o_{26}	u_5	5	C

Table 4.3 – *Data set from table 4.2 in the* observation *representation.*

Variable parts

From the data set in Table 4.2, we know the domain of each variable (with ranks in the case of quantitative variables):

$$\begin{aligned}
\mathcal{V}_1 &= \{1, 2, 3, 4\},\\
\mathcal{V}_2 &= \{1, 2, 3, 4, 5, 6\},\\
\mathcal{V}_3 &= \{1, 2, 3, 4\},\\
\mathcal{V}_4 &= \{a, b, c, d\},\\
\mathcal{V}_5 &= \{A, B, C\}.
\end{aligned}$$

Variable parts are obtained by partitioning those sets, with an ordering constraint for quantitative variables. For instance, suppose the following partitions:

$$\begin{aligned}
P_1 &= \{P_{1,1}, P_{1,2}\} = \{\{1, 2\}, \{3, 4\}\},\\
P_2 &= \{P_{2,1}, P_{2,2}\} = \{\{1, 2, 3, 4\}, \{5, 6\}\},\\
P_3 &= \{P_{3,1}\} = \{\{1, 2, 3, 4\}\},\\
P_4 &= \{P_{4,1}, P_{4,2}, P_{4,3}\} = \{\{a, c\}, \{b\}, \{d\}\},\\
P_5 &= \{P_{5,1}, P_{5,2}\} = \{\{A\}, \{B, C\}\}.
\end{aligned}$$

Using the binary variable representation associated to those variables parts enables us to translate Table 4.2 into Table 4.4.

instance	$X_{1,1}$	$X_{1,2}$	$X_{2,1}$	$X_{2,2}$	$X_{3,1}$	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{5,1}$	$X_{5,2}$
u_1	1	0	1	0	0	1	1	0	1	0
u_2	0	1	1	0	1	0	1	0	0	1
u_3	1	0	0	0	1	1	0	0	1	0
u_4	0	0	0	1	1	1	0	0	0	1
u_5	0	1	0	1	1	0	0	1	0	1

Table 4.4 – *A binary representation of the data based on the variable parts.*

Co-clustering

Co-clustering operates in our model at the *instance* level (i.e., a partition of $\mathcal{U} = \{u_1, \ldots, u_I\}$) and at the *variable part* level (i.e., a partition of $\mathcal{P} = P_1 \cup \ldots \cup P_K$). For data in Table 4.2, and using the previously chosen variable parts, one possible co-clustering structure would be the following

one:

$$\begin{aligned}
G_u &= 2,\\
G_p &= 4,\\
\mathbf{C}^u &= \{\{u_1, u_3\}, \{u_2, u_4, u_5\}\},\\
\mathbf{C}^p &= \{\{P_{1,1}, P_{5,1}\}, \{P_{1,2}, P_{2,2}, P_{4,3}\},\\
&\qquad \{P_{2,1}, P_{4,2}\}, \{P_{3,1}, P_{4,1}, P_{5,2}\}\}.
\end{aligned}$$

An example of a co-cluster is therefore

$$\{u_1, u_3\} \times \{P_{1,1}, P_{5,1}\} = \{u_1, u_3\} \times \{X_1 \in \{1, 2\}, X_5 = A\}.$$

The co-clustering can be summarized by a contingency table that counts the number of observations in each co-cluster, as illustrated in Table 4.5.

	C_1^p $X_1 \in \{1,2\}$, $X_5 = A$	C_2^p $X_1 \in \{3,4\}, X_2 \in \{5,6\}$, $X_4 = d$	C_3^p $X_2 \in \{1,2,3,4\}$, $X_4 = b$	C_4^p $X_3 \in \{1,2,3,4\}$, $X_4 \in \{a,c\}, X_5 \in \{B,C\}$	
$C_1^u = \{u_1, u_3\}$	4	0	2	4	10
$C_2^u = \{u_2, u_4, u_5\}$	0	5	4	7	16
	4	5	6	11	

Table 4.5 – *Contingency table associated to the co-clustering. Each cell contains the number of observations (see Table 4.3) that fulfill the constraints associated to the corresponding clusters: the instance must be in the instance cluster of the row, while the variable must fulfill one of the conditions associated to the variable parts of the column. The last column and row are marginal counts. On this example, one can see that the co-clustering is revealing a dependency between instances and variable parts in the first two columns as some co-clusters are empty.*

Notice that the contingency table itself is the parameter $\mathbf{\Phi}$, i.e.

$$\mathbf{\Phi} = \begin{pmatrix} 4 & 0 & 2 & 4 \\ 0 & 5 & 4 & 7 \end{pmatrix},$$

while the marginal counts $\phi_{g_u \cdot}$ and $\phi_{\cdot g_p}$ given by:

$$\begin{aligned}
\phi_{g_u \cdot} &= (10, 16),\\
\phi_{\cdot g_p} &= (4, 5, 6, 11).
\end{aligned}$$

The counts per instance and per variable part n^u and n^p are given by

$$\begin{aligned}
n^u &= (5, 7, 5, 4, 5),\\
n^p &= (2, 2, 4, 2, 4, 4, 2, 1, 2, 3),
\end{aligned}$$

for the instances $(u_1, \ldots, u_5)$ and the variable parts ($X_{1,1}$, $X_{1,2}$, $X_{2,1}$, $X_{2,1}$, $X_{3,1}$, $X_{4,1}$, $X_{4,2}$, $X_{4,3}$, $X_{5,1}$, $X_{5,2}$), respectively.

Finally, the n^v can be obtained from the data themselves, leading to

$$\begin{aligned} n^v &= (n_4^v, n_5^v), \\ n_4^v &= (2, 2, 2, 1), \\ n_5^v &= (2, 1, 2), \end{aligned}$$

using the ordering given above for $\mathcal{V}_4$ and $\mathcal{V}_5$.

This illustrates the fact that the parameters are redundant and that, when we restrict ourselves to parameters that are compatible with a given data set (as per Definition 1), knowing the co-clustering structure $((P_k)_{1 \leq k \leq K}, \mathbf{C}^u, \mathbf{C}^p)$ is sufficient to determine all the parameters of the model.

4.2.6 Data generation mechanism

Given the parameter Θ, the following hierarchical model is used to generate a set of ϕ observations $\mathcal{O} = \{o_1, \ldots, o_\phi\}$ associated to μ instances forming the set $\mathcal{U} = \{u_1, \ldots, u_\mu\}$. Notice first that the number of observations is given by $\phi = \sum_{i,j} \phi_{i,j}$.

1. The distribution of the ϕ observations over the co-clusters which is defined by a random mapping F from $\{1, \ldots, \phi\}$ to the set of $G = G_u \times G_p$ co-clusters, i.e. to $\{1 \leq g_u \leq G_u\} \times \{1 \leq g_p \leq G_p\}$. The mapping chooses which co-cluster is responsible for generating the corresponding observation: o_l is generated by co-cluster $F(l)$. F is distributed uniformly in the set of compatible mappings. That is, all mappings that respect the counts given in Φ are equally probable.

   Under these constraints, combinatorial arguments show that the probability of such a mapping is

$$P(F = f | G_u, G_p, \phi, \Phi) = \frac{\prod_{g_u=1}^{G_u} \prod_{g_p=1}^{G_p} \phi_{g_u,g_p}!}{\phi!}.$$

   Once a distribution f is chosen, ϕ_{g_u,g_p} (for every g_u and g_p) becomes known. By summation, we can deduce $\phi_{g_u.} = \sum_{g_p=1}^{G_p} \phi_{g_u,g_p}$ and $\phi_{.g_p} = \sum_{g_u=1}^{G_u} \phi_{g_u,g_p}$.

2. On one hand, given F, each cluster of instances $C_{g_u}^u$ is responsible for generating $\phi_{g_u.}$ observations. On the other hand, the partition of the instances to clusters of instances (the model parameter $\mathbf{C}^u$) gives the instances per cluster. The observations in the cluster are generated

by a distribution of the $\phi_{g_u.}$ observations over the instances, which is defined by a mapping $F^u_{g_u}$ from $\{l|o_l \in C^u_{g_u}\}$ to $\{i|u_i \in C^u_{g_u}\}$.

The mapping $F^u_{g_u}$ is distributed uniformly on the set of all compatible mappings, that is mappings that respect the counts given in n^u. Therefore

$$P(F^u_{g_u} = f^u_{g_u}|F = f, \phi, \Phi, n^u, \mathbf{C}^u) = \frac{\prod_{u_i \in C^u_{g_u}} n^u_{g_u,i}!}{\phi_{g_u.}!}.$$

Conditionally on F, mappings for the different clusters of instances are independent random variables. The G_u mappings can thus be collated into a global mapping F^u, from $\{1, \ldots, \phi\}$ to $\mathcal{U}$, such that $o_l = (F^u(l), k_l, v_l)$ for all l. Under these constraints, combinatorial arguments show that the probability of a global mapping f^u is

$$P(F^u = f^u|F = f, \phi, \Phi, n^u, \mathbf{C}^u) = \prod_{g_u=1}^{G_u} \frac{\prod_{u_i \in C^u_{g_u}} n^u_{g_u,i}!}{\phi_{g_u.}!}.$$

Note that the product over all instance clusters and all instances in clusters is identical to the simple product over all instances:

$$\prod_{i=1}^{G_u} \prod_{u_i \in C^u_{g_u}} n^u_{g_u,i}! = \prod_{i=1}^{\mu} n^u_i!.$$

3. Similarly, and independently, we define a random mapping F^p to generate the variable part associated to each observation. For each variable part cluster $C^p_{g_p}$, $F^p_{g_p}$ maps the $\phi_{.g_p}$ observations in the cluster to the variable parts forming the cluster. The distribution is defined by a random mapping from $\{l|o_l \in C^p_{g_p}\}$ to $\{j|P_{k,j} \in C^p_{g_p}\}$, with respect to the counts given in n^p.

   Under the same conditional independence condition used for the instance clusters, F^p is the collated global mapping. We define also $H^p(l)$ as the index of the variable from which $F^p(l)$ is a variable part. Those random mapping are such that $o_l = (u_l, H^p(l), v_l)$ and $v_l \in F^p(l)$.

   A similar uniform probability as in the case of instances is used, which leads to the probability of a mapping f^p:

$$P(F^p = f^p|F = f, \phi, \Phi, \mathbf{C}^p, n^p) = \prod_{g_p=1}^{G_p} \frac{\prod_{P_{k,l} \in C^p_{g_p}} n^p_{g_p,l}!}{\phi_{.g_p}!}.$$

   Note that the product $\prod_{g_p=1}^{G_p} \prod_{P_{k,l} \in C^p_{g_p}}$ over all variable part clusters and all parts in the cluster is equal to the product over all parts for all variables. Thus: $\prod_{g_p=1}^{G_p} \prod_{P_{k,l} \in C^p_{g_p}} n^p_{g_p,l}! = \prod_{k=1}^{K} \prod_{j_k=1}^{J_k} n^p_{g_p,l}!$.

4. Given the variable part associated to each observation (via F^p), we now generate the actual categorical values. For each variable part $P_{k,j}$, a random mapping F_j^c from the set of observations associated to the part $\{l|o_l \in P_{k,j}\}$ to the values in the part $\{t|v_t \in P_{k,j}\}$ is considered. Then for $o_l = (u_l, k_l, v_l)$, $k_l = k$ implies $v_l = F_j^c(l)$. Conditionally on F^p, all F_j^c are independent.

   In the case of qualitative variables, F_j^c is distributed uniformly on the set of mappings that respect the counts given in n^v. Therefore, for any f_j^c,

$$P(F_j^c = f_j^c | F^p = f^p, n_k^p, n^v) = \frac{\prod\limits_{v_t \in P_{k,j}} n_{k,t}^v!}{n_{k,j}^p!}.$$

   The probability of a collated global mapping f^c over all variables is therefore

$$P(F^c = f^c | F^p = f^p, n_k^p, n^v) = \prod_{k \in \mathcal{X}_c} \prod_{P_{k,l} \in P_k} \frac{\prod\limits_{v_t \in P_{k,j}} n_{k,t}^v!}{n_{k,j}^p!}.$$

5. In the case of numerical variables, the mapping F_j^n is a bijection from $\{l|F^p(l) = P_{k,j}\}$ to $P_{k,j}$ (according to equation (4.5)). Using a uniform distribution on those bijections, for any f_j^n

$$P(F_j^n = f_j^n | F^p = f^p, n^p) = \frac{1}{n_{k,j}^p!}.$$

   for the part $P_{k,j}$ of the variable $X_k \in \mathcal{X}_n$.

   The different parts of the same variable now belong to different clusters. Therefore, the mappings are independent. Also, the mappings of the different numerical variables are independent. Therefore, the collated global mapping f^n for all the numerical variables is given by

$$P(F^n = f^n | F^p = f^p, n^p) = \prod_{X_k \in \mathcal{X}_n} \prod_{j=1}^{J_k} \frac{1}{n_{k,j}^p!}.$$

   Denoting $F^k = F^c \cup F^n$, the collated random mappings to the values regardless of the variable type, we have $o_l = (u_l, k_l, v_l)$ with $k_l = H^p(l)$ and $v_l = F^k(l)$.

The final data set is

$$\mathcal{O} = \left\{ \left(F^u(l), H^p(l), F^k(l) \right) \right\}_{1 \leq l \leq \phi}.$$

The generative model. Figure 4.1 provides a simplified representation of the proposed generative model. Parameters are omitted for clarity. Collated maps F^u and F^p, as well as the variable index map H^p are not represented on the graphical model since they are deterministic functions of the random variables $(F_{g_u}^u)_{1 \leq g_u \leq G_u}$ and $(F_{g_p}^p)_{1 \leq g_p \leq G_p}$.

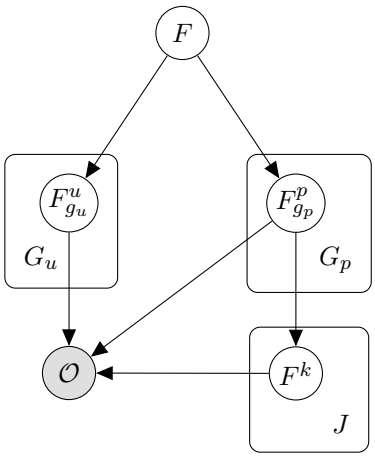


Figure 4.1 – *A directed graphical model of the distribution. Parameters are omitted on this representation for simplicity.*

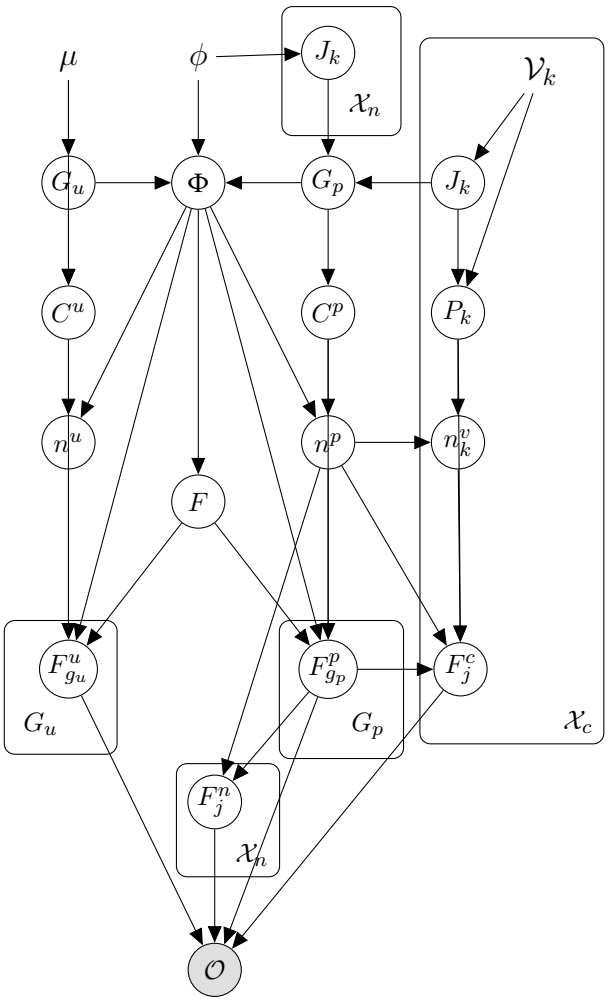


Figure 4.2 – *A directed graphical model of the full distribution. Variable related elements have been separated into one plate for qualitative variables (with $\mathcal{X}_c$ as the range of the plate) and two plates for quantitative variables (with $\mathcal{X}_n$ as the range of the plates).*

Bayesian version of the model. Figure 4.2 gives a representation of the full distribution as a directed graphical model. As in Figure 4.1, derived elements such as collated maps have been omitted for clarity. Notice also that, while variable partitions on quantitative variables have been presented as a parameter of the model, they are derived in a deterministic way from n^p. Thus, they are also omitted from the graphical model.

Likelihood. The likelihood of the data $\mathcal{O}$ generated according to the model defined by Θ is given by the product of probabilities which simplifies to:

$$\mathcal{L}(\Theta|\mathcal{O},\mathcal{U}) = \frac{\left(\prod_{g_u=1}^{G_u}\prod_{g_p=1}^{G_p}\phi_{g_u,g_p}!\right)\left(\prod_{i=1}^{\mu}n_i^u!\right)\left(\prod_{X_k\in\mathcal{X}_c}\prod_{t=1}^{|\mathcal{V}_k|}n_{k,t}^v!\right)}{\phi!\left(\prod_{g_u=1}^{G_u}\phi_{g_u.}!\right)\left(\prod_{g_p=1}^{G_p}\phi_{.g_p}!\right)}. \quad (4.6)$$

4.2.7 Parameter estimation

Given some observed data $\mathcal{O} = \{(o_l, k_l, v_l)\}_{1\leq l\leq N}$, we use a Maximum A Posteriori (MAP) approach to estimate the parameters of a model Θ.

Notations. From the model Θ and the observed data $\mathcal{O} = \{(o_l, k_l, v_l)\}_{1\leq l\leq N}$, let us introduce the following necessary notations.

1. From the data $\mathcal{O}$
   - N: number of observations,
   - I: number of instances,
   - K_n: number of numerical variables,
   - K_c: number of categorical variables,
   - $V_k = |\mathcal{V}_k|$: number of values for a categorical variable X_k,
   - $N_i^u = |\{l|o_l = u_i\}|$: number of observations associated to the instance u_i,
   - $N_{k,t}^v = |\{l|k_l = k \text{ and } v_l = v_t\}|$: number of observations for the value v_t of the categorical variable X_k (with $X_k \in \mathcal{X}_c$ and $v_t \in \mathcal{V}_k$).
2. From the model Θ
   - J_k: number of parts of the k^{th} variable X_k,
   - $J = \sum_k J_k$: number of variable parts,
   - G_u: number of instance clusters,

- G_p: number of variable part clusters,
- $G = G_u \times G_p$: number of co-clusters,
- $m_{g_u}^{(u)}$: number of instances in the g_u^{th} cluster of instances,
- $m_{g_p}^{(p)}$: number of variable parts in the g_p^{th} cluster of variable parts,
- $m_{j_k}^{p}$: number of values in the j_k^{th} part of X_k.

3. From the couple $(\mathcal{O}, \Theta)$

- $N_{g_u,g_p} = \left|\left\{ l \middle| o_l \in C_i^u \text{ and } v_l \in \bigcup_{V \in C_j^p} V \right\}\right|$: number of observations (from the data) that belong to the co-cluster formed by the g_u^{th} cluster of instances and the g_p^{th} cluster of variable parts,
- $N_{g_u}^{(u)} = \sum_{g_p} N_{g_u,g_p}$: number of observations (from the data) that belong to the g_u^{th} cluster of instances (with $g_u \in 1 \dots G_u$),
- $N_{g_p}^{(p)} = \sum_{g_u} N_{g_u,g_p}$: number of observations (from the data) that belong to the g_p^{th} cluster of variable parts (with $g_p \in 1 \dots G_p$),
- $N_{k,j_k}^{p} = |\{l | k_l = k \text{ and } v_l \in P_{k,j_k}\}|$: number of observations (from the data) that belong to the j_k^{th} part of X_k (with $j_k = 1 \dots J_k$).

For some observed data $\mathcal{O}$ and a model Θ, we define the following compatibility constraint.

Definition 1 *Let $\mathcal{O} = \{(o_l, k_l, v_l)\}_{1 \leq l \leq N}$ be some observed data containing N observations, associated to I instances $\mathcal{U} = \{u_1, \dots, u_I\}$. A parameter*

$$\Theta = (\mu, (J_k, P_k, n_k^v)_{1 \leq k \leq K}, G_u, \mathbf{C}^u, G_p, \mathbf{C}^p, \phi, n^u, n^p)$$

is compatible with the observed data $\mathcal{O}$ and $\mathcal{U}$ if and only if:

1. $\mu = I$,
2. $\phi = N$,
3. $\forall i,\ n_i^u = N_i^u$,
4. $\forall k, j_k,\ n_{k,j_k}^p = N_{k,j_k}^p$,
5. $\forall k | X_k \in \mathcal{X}_c,\ \forall t | v_t \in \mathcal{V}_k,\ n_{k,t}^v = N_{k,t}^v$,
6. $\forall (g_u, g_p),\ \phi_{g_u,g_p} = N_{g_u,g_p}$,

where: $1 \leq i \leq I$, $1 \leq k \leq K$, $1 \leq j_k \leq J_k$, $1 \leq t \leq |\mathcal{V}_k|$, $1 \leq g_u \leq G_u$, and $1 \leq g_p \leq G_p$.

The first condition implies that the model parameter for the number of instances is equal to the observed number of instances. The second condition implies that the model parameter for the total number of observations is equal to the observed number. The third, fourth and fifth conditions state that the number of observations per instance, per variable part, and per categorical value, in the model, are all equal to those in the observed data. The last condition implies that the counts per co-cluster coincide with the data and that $N = \sum_{g_u,g_p} \phi_{g_u,g_p}$.

The likelihood is considered null when the model Θ is not *compatible* with the data $\mathcal{O}$. When it is compatible, the model parameters become directly related to the corresponding counts in the data set via the clustering parameters and the compatibility restrictions. Therefore, the likelihood of the observed data can be written directly using these counts:

$$\mathcal{L}(\Theta|\mathcal{O},\mathcal{U}) = \frac{\left(\prod_{g_u=1}^{G_u}\prod_{g_p=1}^{G_p} N_{g_u,g_p}!\right)\left(\prod_{i=1}^{I} N_i^u!\right)\left(\prod_{X_k\in\mathcal{X}_c}\prod_{t=1}^{|\mathcal{V}_k|} N_{k,t}^v!\right)}{N!\left(\prod_{i=1}^{G_u} N_{g_u}^{(u)}!\right)\left(\prod_{g_p=1}^{G_p} N_{g_p}^{(p)}!\right)}. \quad (4.7)$$

Prior distribution on the parameters

As mentioned earlier, the model is defined using a hierarchy of the parameters where, at each stage of the hierarchy, the parameters are chosen with respect to the previous ones. We use a non informative prior on the parameters to perform a Maximum A Posteriori (MAP) estimation. The prior distribution is built hierarchically, using uniform distributions at each level. This enables us to obtain a co-clustering model automatically without user input while letting the data "speak for itself".

Globally, the model parameters are defined by 3 levels of hierarchy as follows.

1. The variable partitions:

   (a) The number of parts per variable.

   - For a categorical variable $X_k \in \mathcal{X}_c$, the number of parts is uniformly distributed between the 1 and the number of values of the variable $V_k = |\mathcal{V}_k|$. The probability of choosing a given number of parts J_k is therefore:

   $$P(J_k) = \frac{1}{V_k}, \text{ for } k|X_k \in \mathcal{X}_c. \quad (4.8)$$

   - For a numerical variable, the number of parts is uniformly distributed on the values between 1 and the total number of

observations N. The probability of choosing a given number of parts J_k is therefore:

$$P(J_k) = \frac{1}{N}, \text{ for } k \mid X_k \in \mathcal{X}_n. \tag{4.9}$$

Knowing the number of parts per variable, the total number of parts J is given by: $J = \sum_{k=1}^{K} J_k$.

(b) The partition of the values.

- For a categorical variable X_k, all partitions of its V_k values into the previously chosen number of parts J_k are equally likely. The number of possible ways of partitioning V_k values into J_k groups is given by the sum of Stirling numbers of the second kind $\sum_{j_k=1}^{J_k} S(V_k, j_k)$. Hence, the probability of choosing a particular partition P_k is:

$$P(P_k|J_k) = \frac{1}{B(V_k, J_k)}, \text{ for } k \mid X_k \in \mathcal{X}_c. \tag{4.10}$$

where $B(V_k, J_k)$ is the sum of Stirling numbers of the second kind $B(V_k, J_k) = \sum_{j_k=1}^{J_k} S(V_k, j_k)$.
For each categorical variable $X_k \in \mathcal{X}_c$, we can now deduce the number $m^p_{j_k}$ of values per part j_k with $j_k \in \{1 \dots J_k\}$.

- Because of the constraints of contiguous intervals on the quantitative variables, knowing the number of intervals J_k and the number of observations per part N^p_{k,j_k} is enough to know the partition. Therefore, we chose to build a prior on N^p_{k,j_k} rather than on the partitions.

2. The co-clustering structure:

(a) The number of instance clusters is uniformly distributed on the values between 1 and I, the number of instances. Similarly, and independently, the number of variable part clusters is uniformly distributed between 1 and J, the number of variable parts. The probabilities of choosing a given number G_u of instance clusters and a given number G_p of variable part clusters are given by:

$$P(G_u) = \frac{1}{I} \text{ and } P(G_p) = \frac{1}{J}. \tag{4.11}$$

(b) All partitions of the I instances into G_u clusters are equally probable. The probability of choosing a given partition $\mathbf{C}^u$ is therefore:

$$P(\mathbf{C}^u|G_u) = \frac{1}{B(I, G_u)}, \tag{4.12}$$

where $B(I, G_u)$ is, as defined above, the sum of Stirling numbers of the second kind $B(I, G_u) = \sum_{g_u=1}^{G_u} S(I, g_u)$ giving the number of ways to partition I into G_u clusters.

From the chosen partition $\mathbf{C}^u$, we can now deduce the number $m_{g_u}^{(u)}$ of instances per instance cluster.

(c) Similarly to the partition of instances, all partitions of the J variable parts into G_p clusters are equally probable. The probability of choosing a given partition $\mathbf{C}^p$ is therefore:

$$P(\mathbf{C}^p|G_p) = \frac{1}{B(J, G_p)}, \tag{4.13}$$

where $B(J, G_p)$ is as defined above.

Notice that $\mathbf{C}^p$ is defined on the indexes of the variable parts within the set $\mathcal{P} = P_1 \cup \ldots \cup P_K$ rather than on their actual content. This allows to postpone the definition of the contents of those variable parts for the quantitative variables (as they are obtained through n^p).

From the chosen partition $\mathbf{C}^p$, we can now deduce the number $m_{g_p}^{(p)}$ of parts per variable part cluster.

3. The actual clustering of the observations, which is defined by three levels of multinomial distributions as follows.

(a) Distribution of all the N observations over the co-clusters. All possible distributions of the N observations over the $G = G_u \times G_p$ co-clusters are equally probable. In other words, the matrix of counts $\mathbf{N} = (N_{g_u,g_p})_{1 \leq g_u \leq G_u, 1 \leq g_p \leq G_p}$ is distributed uniformly in the set of integer valued $(G_u \times G_p)$-matrices whose contents sum to N. The number of such matrices is given by $\binom{N+G-1}{G-1}$. Therefore

$$P(\mathbf{N}|\ G) = \frac{1}{\binom{N+G-1}{G-1}}. \tag{4.14}$$

Once a matrix $\mathbf{N}$ is chosen, N_{g_u,g_p} (for every g_u and g_p) becomes known. By summation, we deduce $N_{g_u}^{(u)} = \sum_{g_p=1}^{G_p} N_{g_u,g_p}$ and $N_{g_p}^{(p)} = \sum_{g_u=1}^{G_u} N_{g_u,g_p}$.

(b) Distributing the observations of a cluster over instances/parts within the cluster.

i. All distributions of the $N_{g_u}^{(u)}$ observations over the $m_{g_u}^{(u)}$ instances of the g_u^{th} instance cluster that fulfill the structural

constraint in equation (4.2) are equally probable. The number of possible distributions is $\binom{N_{g_u}^{(u)} + m_{g_u}^{(u)} - 1}{m_{g_u}^{(u)} - 1}$ for each cluster of instances $C_{g_u}^u$. The probability of choosing one of these distributions is therefore

$$P(\{N_i^u\}_{\{i \in g_u\}} \mid G, \mathbf{N}) = \frac{1}{\binom{N_{g_u}^{(u)} + m_{g_u}^{(u)} - 1}{m_{g_u}^{(u)} - 1}}.$$

Conditionally on the clustering of instances, these distributions are independent. Therefore the probability of distributing all observations on all instances is

$$P(\mathbf{N}^u \mid G, \mathbf{N}) = \prod_{g_u=1}^{G_u} \frac{1}{\binom{N_{g_u}^{(u)} + m_{g_u}^{(u)} - 1}{m_{g_u}^{(u)} - 1}}, \tag{4.15}$$

where $\mathbf{N}^u$ is the vector containing all the counts.

ii. Similarly, but independently, all distributions of the $N_{g_p}^{(p)}$ observations over the $m_{g_p}^{(p)}$ variable parts of the part cluster $C_{g_p}^p$ that fulfill the structural constraint in equation (4.3) are equally probable. The number of such distributions is $\binom{N_{g_p}^{(p)} + m_{g_p}^{(p)} - 1}{m_{g_p}^{(p)} - 1}$.
Thus, the probability of distributing all observations over all variable parts is given by:

$$P(\mathbf{N}^p \mid G, \mathbf{N}) = \prod_{g_p=1}^{G_p} \frac{1}{\binom{N_{g_p}^{(p)} + m_{g_p}^{(p)} - 1}{m_{g_p}^{(p)} - 1}}, \tag{4.16}$$

where $\mathbf{N}^p = (N_1^p, \ldots, N_J^p)$ is the vector containing all the counts of number of observations per part.

(c) Distributing the observations of a variable part over the values in the part.

i. For every cluster of variable parts $C_{g_p}^p$ and every categorical part $j_k \in C_{g_p}^p$, all distributions of the N_{k,j_k}^p observations in the part over the set of $m_{j_k}^p$ values that constitute the variable part are equally probable, under the structural constraint in equation (4.4). The probability of choosing one distributing is

$$P(\mathbf{N}^v \mid G, \mathbf{N}, \mathbf{N}^p, g_p) = \prod_{j_k \in C_{g_p}^p \mid X_k \in \mathcal{X}_c} \frac{1}{\binom{N_{k,j_k}^p + m_{j_k}^p - 1}{m_{j_k}^p - 1}},$$

for a given cluster $C^p_{g_p}$.
Given a clustering $\mathbf{C}^p$, the distributions are independent. Hence, the probability of distributing all the observations is given by

$$P(\mathbf{N}^v|\, G, \mathbf{N}, \mathbf{N}^p) = \prod_{g_p=1}^{G_p} \prod_{j_k \in C^p_{g_p} \,|\, X_k \in \mathcal{X}_c} \frac{1}{\binom{N^p_{k,j_k} + m^p_{j_k} - 1}{m^p_{j_k} - 1}}, \tag{4.17}$$

where $\mathbf{N}^v$ is the vector containing all the counts of number of observations for the categorical values. However, this is equivalent to distributing all the observations per categorical variable over the parts of the variable. Therefore equation (4.17) is equivalent to the following equation (4.18), which we will prefer for its simplicity:

$$P(\mathbf{N}^v|\, G, \mathbf{N}, \mathbf{N}^p) = \prod_{X_k \in \mathcal{X}_c} \prod_{j_k=1}^{J_k} \frac{1}{\binom{N^p_{k,j_k} + m^p_{j_k} - 1}{m^p_{j_k} - 1}}. \tag{4.18}$$

ii. In the case of numerical variables, the number of observations N^p_{k,j_k} per variable part (given by the mapping of the observations of the variable part cluster on the parts) indicates the boundaries of the intervals. Because the parts are ordered, finding the correct ranks of the observations within each interval gives the global ranking of the observations in the variable, which is a likelihood term.

From the equations (4.8) to (4.18), the prior probability of the model parameters $\mathcal{M}$ (written in terms of the data counts) is given by

$$\begin{aligned} P(\mathcal{M}) = & \prod_{k=1}^{K_n} P(J_k|X_k \in \mathcal{X}_n) \prod_{k=1}^{K_c} P(J_k|X_k \in \mathcal{X}_c) P(P_k|J_k, X_k \in \mathcal{X}_c) \\ & P(G_u|I) P(\mathbf{C}^u|G_u, I) P(G_p|J) P(\mathbf{C}^p|G_p, J) \\ & P(\mathbf{N}|G_u, G_p) \\ & P(\mathbf{N}^u\,|G_u, G_p, \mathbf{N}, \mathbf{C}^u) P(\mathbf{N}^p|G_u, G_p, \mathbf{N}, \mathbf{C}^p) P(\mathbf{N}^v|G_u, G_p, \mathbf{N}, \mathbf{C}^p, \mathbf{N}^p). \end{aligned} \tag{4.19}$$

MAP based model selection criterion

The product of the prior distributions (equation (4.19)) and the likelihood (equation (4.7)) results in the posterior probability, the negative log of which is used to build the criterion given by Defintion 2.

Definition 2 *According to the MAP approach, the best parameters are the ones that minimize the following criterion:*

$$\begin{aligned}\mathcal{C}(\mathcal{M}) = &\sum_{X_k \in \mathcal{X}_c} \log V_k + K_n \log N + \sum_{X_k \in \mathcal{X}_c} \log B(V_k, J_k) \\ &+ \log I + \log J + \log B(I, G_u) + \log B(J, G_p) + \log \binom{N+G-1}{G-1} \\ &+ \sum_{g_u=1}^{G_u} \log \binom{N_{g_u}^{(u)} + m_{g_u}^{(u)} - 1}{m_{g_u}^{(u)} - 1} + \sum_{g_p=1}^{G_p} \log \binom{N_{g_p}^{(p)} + m_{g_p}^{(p)} - 1}{m_{g_p}^{(p)} - 1} \\ &+ \sum_{X_k \in \mathcal{X}_c} \sum_{j_k=1}^{J_k} \log \binom{N_{k,j_k}^{p} + m_{j_k}^{p} - 1}{m_{j_k}^{p} - 1} + \log N! - \sum_{g_u=1}^{G_u} \sum_{g_p=1}^{G_p} \log N_{g_u,g_p}! \\ &+ \sum_{g_u=1}^{G_u} \log N_{g_u}^{(u)}! - \sum_{i=1}^{I} \log N_i^u! + \sum_{g_p=1}^{G_p} \log N_{g_p}^{(p)}! - \sum_{X_k \in \mathcal{X}_c} \sum_{t=1}^{V_k} \log N_{k,t}^{v}!.\end{aligned} \tag{4.20}$$

Interpretation. The criterion (4.20) contains terms that correspond to the prior distribution of the parameters and terms that come from the likelihood of the data given the parameters. The likelihood terms tend to favor complex models that fit well the data, whereas the prior terms, that increase with the number of parameters, have a regularization role and tend to favor simpler models. Hence, this cost function is a discrete and regularized model selection criterion that performs a trade off between the goodness of fit of the model (given by the likelihood part) and the model complexity evaluated by the prior related part. By optimizing this criterion, the instances similarly distributed on the clusters of variable parts are grouped together and, inversely, the variable parts similarly distributed on the clusters of instances are grouped together. Furthermore, the further the optimization is pushed, the better is the resulting model, yet the criterion will not allow over-fitting because it is regularized by the prior cost.

The proposed model is thus a MAP data descriptive approach to transform variables into dense categories (parts) and perform a simultaneous grouping of the objects and the new categories. Given the optimized variable partitions, the simultaneous clustering is MAP optimum. However, the variable parts are optimized with respect to a simultaneous clustering of the instances and of the resulting parts. It is thus a recursive cycle that should converge to an optimal trade off.

Optimization strategy

The criterion given in equation (4.20) is a non convex discrete criterion which is quite difficult to optimize. Fortunately, the criterion is close to the one used in the MODL approach (Boullé 2011) (see also Guigourès et al. (2015b)). In particular, when the variable parts are fixed, the proposed approach can be seen as a direct application of the MODL approach to

the binary variables associated to variable parts. Thus, a natural solution consists in leveraging solutions developed for the MODL approach which are already implemented in the software Khiops[1]. More precisely, the criterion is optimized as follows.

1. Initialization: choose a set of initial numbers of parts $\mathbf{J}$.

   (a) Initial variable parts: for each $j \in \mathbf{J}$, create an initial partition of the variables into j variable parts. For numerical variables, those partitions are uniform (in term of frequency, as implied by the rank representation). For qualitative variables, the most frequent values are kept in separated variable parts while the last variable part gathers the less frequent values.

   (b) Initial co-clusters: MODL co-clustering is applied on the binary variables obtained from the previous step. The obtained co-clustering is considered as an initial solution.

   These two steps give a set of initial partitions and co-clustering structures.

2. Evaluate each of the initial structures with respect to the criterion in equation (4.20) and retain the best one as the starting point of a model (and partition) refining process.

3. Model refining: the chosen model is refined using an iterative greedy moving/merging process. The process consists of iterative testing of the effect (on the criterion) of operations such as moving a variable part from a cluster to another, moving a value from a variable part to another, merging clusters, merging variable parts, etc. Operations that decrease the criterion are accepted. When no further improvement can be done, the obtained co-clustering structure is considered as the *optimal model*. It is clear that the this process provides only locally optimum solutions.

Null model. The null model is when all variables are partitioned into one single interval or group and there is only one co-cluster ($G = G_u = G_p = J_k = 1$).

Definition 3 *According to the MAP approach, the criterion of a null model $\mathcal{M}_\emptyset$ is given*

[1] `http://www.khiops.com/`

by:

$$\begin{aligned}\mathcal{C}(\mathcal{M}_\emptyset) = & \sum_{X_k \in \mathbf{X}_c} \log V_k + K_n \log N + \log I + \log K \\ & + \log \binom{N+I-1}{I-1} + \log \binom{N+K-1}{K-1} \\ & + \sum_{X_k \in \mathbf{X}_c} \log \binom{n_{.k}+V_k-1}{V_k-1} + 2 \log N! \\ & - \sum_{i=1}^{I} \log (n_{i.}!) - \sum_{X_k \in \mathbf{X}_c} \sum_{v_k=1}^{V_k} \log (n_{v_k}!)\end{aligned} \tag{4.21}$$

This criterion corresponds to the posterior probability of distributing the N observations over the K variables, over the I instances, and over the values of each categorical variable regardless of any partitioning.

Model coarsening for interpretation

The *optimal model* is considered an optimal starting point for the exploratory analysis. However, when the data is complex and large, this optimal co-clustering remains very detailed and complex for easy exploitation. Thus, to ease the analysis of the data using the model, like in Section 3.3.1, a process of coarsening the clusters is used. Like in Section 3.3.1, we use a greedy procedure that chooses automatically the best dimension to merge between instances and variable parts at each step. Furthermore, given that the criterion (equation 4.20) gives the exact posterior probability of the model, we can compute the degradation in probability when using the coarser model (as per Definition 4).

Definition 4 *Let $\mathcal{M}$ be a co-clustering model, we define a similarity between two clusters of the same type (two clusters of instances or two clusters of variable parts) C_1 and C_2 as follows:*

$$S(C_1, C_2) = \mathcal{C}(\mathcal{M}_{C_1 \cup C_2}) - \mathcal{C}(\mathcal{M}), \tag{4.22}$$

where $\mathcal{M}_{C_1 \cup C_2}$ is the obtained model after merging C_1 and C_2.

To define a stopping criteria for the coarsening process, we define a measure of informativeness of the co-clustering that corresponds to the percentage of informativity the co-clustering has kept after the merges, compared to the optimal model and to the null model.

Definition 5 *Let $\mathcal{M}_*$ and $\mathcal{M}_\emptyset$ be the best model as obtained by optimizing (4.20) and the null model. The informativity of a co-clustering $\mathcal{M}$ is given by:*

$$\tau(\mathcal{M}) = \frac{\mathcal{C}(\mathcal{M}) - \mathcal{C}(\mathcal{M}_\emptyset)}{\mathcal{C}(\mathcal{M}_*) - \mathcal{C}(\mathcal{M}_\emptyset)} \tag{4.23}$$

The informativity of a null model is null while the maximum informativity level is considered for the optimized model $\mathcal{M}_*$. The index is bounded $0 \leq \tau(\mathcal{M}) \leq 1$. Hence, all the intermediate models are more probable than the null model and less probable than the best model, by definition.

4.3 Conclusion

In this chapter, we proposed a co-clustering model that applies to data with mixed type variables. While most co-clustering approaches perform a co-clustering via a mapping of the instances to clusters of instances coupled with a mapping of the variables to variable clusters, or via a direct mapping of the observations to co-clusters, our approach performs a mapping of the instances to clusters of instances, a mapping of the variables to variable parts (intervals or groups of values), a mapping of the variable parts to clusters of variable parts, and a mapping of the observations to co-clusters. These mapping have the advantage of allowing us to handle mixed type variables, missing observations and set-valued variables. Furthermore, the model requires no user-defined parameter as the mappings are optimized within the model, using a Maximum A Posteriori approach.

In the next chapter, we show the contribution of this model and contrast its results with those obtained from the methodology proposed in the previous chapter. In particular, the co-clustering model optimizes the numbers of variable parts, the variable partitions, the number of clusters and the clusterings.

Chapter 5

Experimental results

In the previous chapter, we have proposed a parameter-less co-clustering model for mixed type data. In this chapter, we apply the model to real and artificial data sets to illustrate its efficacy and to emphasize its contribution compared to the co-clustering methodology proposed in Chapter 3.

This chapter is organized as follows. Section 5.1 introduces the objectives and motivations of this chapter. Section 5.2 presents experimental results on real-world data sets with increasing sizes and complexities. Section 5.3 illustrates the results of the co-clustering model in extreme cases, namely in the case of independent variables and in the case of perfectly correlated ones, while comparing with the Crosscat model (Mansinghka et al. 2016). Finally, conclusions and axis of improvements are discussed in Section 5.4.

5.1 Introduction

In this chapter, we apply the co-clustering model, proposed in Chapter 4, to data sets of increasing sizes and complexities. To highlight the main features of the co-clustering and compare it with the methodology proposed in Chapter 3, we explore the results using the tools introduced in Section 3.3. In particular, we show that the inferred variable partitions do approximate the global distribution of the variables better than in Chapter 3. Namely, in the methodology proposed in Chapter 3, the number of parts is equal for all variables, the variable parts have equal frequencies, and the origin of the variable part is not leveraged in the sense that a cluster of variable parts may contain many parts of the same variable. Here, we show that when applying the co-clustering model, the number of parts may be different from one variable to another and they do not necessarily have equal frequencies. Instead, the retained partitions are those that are more relevant to the task of co-clustering. In terms of clusters, we show that because the variable partitions are optimized, the clusterings are better tuned and the interpretation of the different clusters can be significantly more precise than in the parameter-based methodology.

To compare with the methodology of Chapter 3, we apply the co-clustering model to the same real-world data sets (Section 3.5). Further-

more, to evaluate the scalability of the co-clustering model, we apply it to an additional large data set containing around 12 millions observations. Keep in mind that the same co-clustering analysis tools explained in Chapter 3 (Section 3.3) apply to the formalized co-clustering model proposed in Chapter 4. Therefore, given a co-clustering, whether optimum or simplified, we use these tools without redefining them.

Additionally to comparing with the proposed methodology, we also compare the results with those obtained using the CrossCat model (described in Section 2.5.3) which is, to the best of our knowledge, the most comparable model. Recall here that CrossCat is a fully Bayesian non parametric approach for density estimation. It uses approximate Bayesian inference to extract a hierarchical structure in which an outer clustering groups the variables in a set of non-overlapping views of independent variables. Then, each view is clustered independently at the level of instances using a separate Dirichlet process mixture.

Notice that CrossCat has a complexity in $\mathcal{O}(IK\tau\sigma)$ where τ is the maximum number of variable clusters (a.k.a. views) and σ the maximum number of instance clusters (see Section 2.5.3). Thus, the scalability of CrossCat and our approach are comparable. Both models can handle millions of observations.

5.2 Experiments on real-world data sets

To contrast the results of the parameter-less co-clustering model with those obtained when applying the approach in Chapter 3, this section explores the results of the model on the data sets Iris and Adult. We also show that the model scales to large data sets. For all these experiments, we follow the optimization procedure explained in Section 4.2.7.

Interpretation methodology

In this section, we proceed as follows for cluster interpretation.

1. **Choose the co-clustering**: perform model coarsening if necessary. Once the desired coarsening level is reached, we compute the matrix of mutual information. For comparison with the methodology (Chapter 3), we give the composition of the co-clusters and examples of the variable partitions.

2. **Choose the clusters to analyze**: select the clusters of instances with the highest mutual information. For each of these clusters, we select the variable part clusters that contribute the most to the information of the cluster, and extract the compositions of the selected clusters for analysis.

3. **Analyze the results**: the selected clusters of instances are characterized by the variable parts contained in their most contributing clusters of variable parts.

4. **Compare the results**: for the data sets already analyzed in Chapter 3, we compare the results with those obtained using the parameter-based approach to illustrate the contribution of the co-clustering model. When possible, the results are also compared with those obtained using CrossCat.

This process can be used iteratively at each level of the hierarchy of co-clusters.

5.2.1 Iris data

Fisher's Iris data (Fisher 1936) is available from the UCI Machine Learning Repository (Lichman 2013) and it consists of 150 instances, 750 observations, 4 numerical variables (*PetalLength*, *PetalWidth*, *SepalLength*, and *SepalWidth*) and 1 categorical *Class* variable (see Section 3.5.1).

Initialization

Following the optimization strategy in 4.2.7, we start by partitioning all variables using predefined partition sizes ranging from 2 to 10 parts per variable. The MODL approach is then applied to the resulting partitioned data to obtain an initial co-clustering structure. We then evaluate the resulting models using the proposed criterion in equation (4.20). Figure 5.1 shows the criterion values per initial model and the optimized model. Before optimization (the red line in Figure 5.1), we find that the minimal criterion value (best seen co-clustering model) is found when each variable is partitioned into 3 parts with roughly equal frequencies, which corresponds to 15 variable parts.

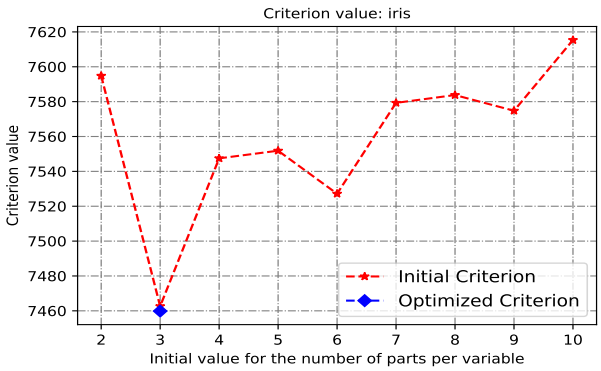


Figure 5.1 – *Iris: evolution of the criterion values.*

The co-clustering

The optimized co-clustering contains 3 clusters of instances, 14 variable parts, and 7 clusters of variable parts, which is easily exploitable. Therefore, no model coarsening is required and we will analyze the optimal co-clustering as it is.

Table 5.1 shows the table of counts containing the number of observations per co-cluster, along with the marginal counts per cluster. Table 5.2 illustrates the co-clustering in terms of mutual information while Figure 5.2 shows a color coded version of the mutual information explained by the optimized co-clustering. In this representation, the red color marks an over representation of the observations compared to the case where the two dimensions would be independent (high values of mutual information). The blue color marks an under representation (low values of mutual information), and the white color represents an empty co-cluster.

Cluster	C_1^p	C_2^p	C_3^p	C_4^p	C_5^p	C_6^p	C_7^p	$\phi_{g_u.}$
C_1^u	150	45	37	0	0	0	18	250
C_2^u	0	1	13	145	38	8	50	255
C_3^u	0	6	5	4	13	143	74	245
$\phi_{.g_p}$	150	52	55	149	51	151	142	$\phi = 750$

Table 5.1 – *Iris: number of observations per co-cluster.*

Cluster	C_1^p	C_2^p	C_3^p	C_4^p	C_5^p	C_6^p	C_7^p	*marginal*
C_1^u	0.219	0.057	0.034	0	0	0	-0.023	0.287
C_2^u	0	-0.003	-0.006	0.203	0.039	-0.019	0.002	0.216
C_3^u	0	-0.008	-0.008	-0.013	-0.004	0.203	0.046	0.216
marginal	0.219	0.046	0.02	0.19	0.035	0.184	0.025	0.719

Table 5.2 – *Iris: mutual information.*

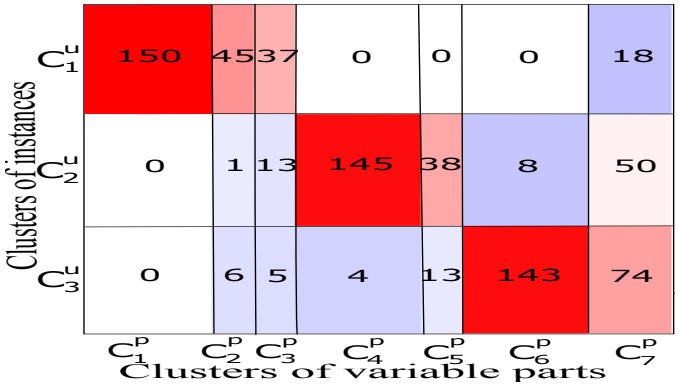


Figure 5.2 – *Iris: the resulting co-clustering.*

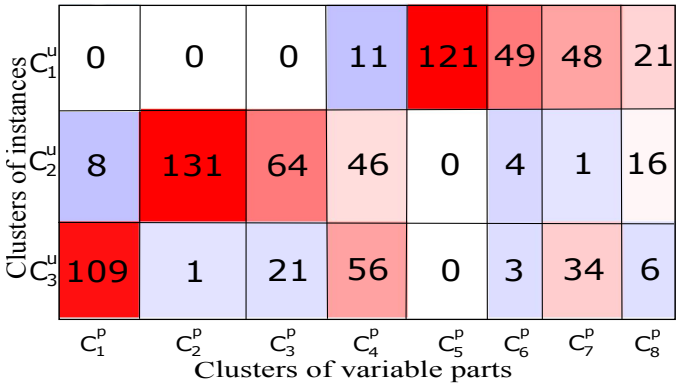


Figure 5.3 – *Iris: the co-clustering resulting from the methodology in Chapter 3.*

For comparison with the parameter-based methodology in Chapter 3 (Section 3.5.1), Figure 5.3 shows the co-clustering resulting from the methodology. Notice that the mutual information of the optimized co-clustering (Table 5.2) provides stronger association between the clusters and a cleaner separation between the different clusters compared to Table 3.4 (Section 3.5.1), as can be seen from the most informative co-clusters (**0.177** against **0.219**, **0.177** against **0.203**, and **0.161** against **0.203**).

The variable parts

On the level of individual variables, the optimum variable partition is given by Table 5.3.

Variable	Parts	n^p
PetalLength	$P_{1,1} = PetalLength]-\infty; 2.4]$	50
	$P_{1,2} = PetalLength]2.4; 4.85]$	49
	$P_{1,3} = PetalLength]4.85; +\infty[$	51
PetalWidth	$P_{2,1} = PetalWidth]-\infty; 0.8]$	50
	$P_{2,2} = PetalWidth]0.8; 1.65]$	52
	$P_{2,3} = PetalWidth]1.65; +\infty[$	48
SepalLength	$P_{3,1} = SepalLength]-\infty; 5.45]$	52
	$P_{3,2} = SepalLength]5.45; 6.25]$	47
	$P_{3,3} = SepalLength]6.25; +\infty[$	51
SepalWidth	$P_{4,1} = SepalWidth]-\infty; 3.15]$	95
	$P_{4,2} = SepalWidth]3.15; +\infty[$	55
Class	$P_{5,1} = Class\{setosa\}$	50
	$P_{5,2} = Class\{virginica\}$	50
	$P_{5,3} = Class\{versicolor\}$	50

Variable	Part	Values	n^v
Class	$P_{5,1}$	*setosa*	50
	$P_{5,2}$	*virginica*	50
	$P_{5,3}$	*versicolor*	50

Table 5.3 – *Iris: the variable parts and their compositions.*

Notice that, compared to the initial co-clustering structure which had 15 parts, two parts of the variable *SepalWidth* have been merged which changed the partitioning and the resulting co-clustering structure. Thus, the variable parts have neither equal numbers of parts nor equal frequencies (Table 5.3).

The clusters

We notice that, for this data set, all clusters of instances have very comparable contribution to the mutual information. Thus, we will explain all of them.

Instance cluster	$\lvert C^u_{g_u}\rvert$
C^u_1	50
C^u_2	51
C^u_3	49

Part cluster	Composition	n^p_j
C^p_1	$Class\{setosa\}$	50
	$PetalLength]-\infty;2.4]$	50
	$PetalWidth]-\infty;0.8]$	50
C^p_2	$SepalLength]-\infty;5.45]$	52
C^p_3	$SepalWidth]3.15;+\infty[$	55
C^p_4	$Class\{virginica\}$	50
	$PetalWidth]1.65;+\infty[$	48
	$PetalLength]4.85;+\infty[$	51
C^p_5	$SepalLength]6.25;+\infty[$	51
C^p_6	$Class\{versicolor\}$	50
	$PetalWidth]0.8;1.65]$	52
	$PetalLength]2.4;4.85]$	49
C^p_7	$SepalWidth]-\infty;3.15]$	95
	$SepalLength]5.45;6.25]$	47

Table 5.4 – *Iris: composition of the clusters.*

Table 5.4 gives the compositions of the clusters of instances and of variable parts.

Analysis of the results

From the extracted co-clusters (Figure 5.2 and Table 5.2), and from the compositions of the clusters (Tables 5.4 and 5.3), it is clear that the resulting co-clustering distinguishes:

- a cluster (C^u_1) of 50 instances containing small flowers characterized by C^p_1 (i.e. $PetalLength]-\infty;2.4]$, $PetalWidth]-\infty;0.8]$, and $Class\{setosa\}$),
- a cluster (C^u_2) of 51 instances containing the large flowers characterized by C^p_4 (i.e. $PetalLength]4.85;+\infty[$, $PetalWidth]1.65;+\infty[$, and $Class\{virginica\}$),
- a cluster (C^u_3) of 49 instances containing the medium sized flowers characterized by C^p_6 (i.e. $PetalLength]2.4;4.85]$, $PetalWidth]0.8;1.65]$, and $Class\{versicolor\}$).

This indicates that the small flowers are *setosa*, the large ones are mostly *virginica*, and the medium sized ones correspond to *versicolor*. We also find that the clusters of instances are characterized mainly by the low, medium and large values for the *PetalLength* and *PetalWidth* variables, and by the variable *Class*, which means that these are the most informative variables

with respect to the clusters of instances. Hence, we recover some well known facts of the Iris data set, such as the correlation between the Petal variables and the fact that these variables are strong predictors for the class variable. Additionally, from the counts in Table 5.1 (second and third clusters of instances), one can deduce a mixture between C_2^u and C_3^u. In fact, knowing the actual nature of the data set, we find that the distribution of the values of at least one versicolor flower is more similar to the virginica distribution than to the other versicolors.

If we are now to consider the composition of the least informative clusters of variable parts, we can conclude the following:

- The cluster C_2^p contains the small *SepalLength* values, and these are mostly present in the cluster of instances C_1^u. The cluster C_3^p contains large *SepalWidth* values, and these values are also mostly present in the cluster of instances C_1^u. From these two clusters, one can characterize the *setosa* flowers (cluster C_1^u) by: small *PetalLength*, small *PetalWidth*, small *SepalLength* and large *SepalWidth*.

- The cluster C_5^p contains the large values of *SepalLength*, and it is mostly present in the cluster of instances C_2^u. Thus, we can characterize the *virginica* flowers (cluster C_2^u) by: large *PetalLength*, large *PetalWidth*, and large *SepalLength*.

- The cluster C_7^p contains medium values of *SepalLength* and small values of *SepalWidth*. This cluster is not very informative as it is present in all the clusters of instances, more in C_3^u than in C_1^u though.

Comparison with the co-clustering methodology

In comparison with the approach proposed in Chapter 3, we notice that globally, the interpretations of the clusters are similar. Both approaches extract three clusters of small, medium and large flowers described by the variables *PetalLength*, *PetalWidth*, and *Class*, which are the most informative for the task of clustering. However, the two approaches are different in many aspects.

1. **Optimized partitioning of the variables.** In the formalized model, the number of variable parts and the partition of the variables are no longer a user parameter. They are optimized automatically as a model parameter. In the first approach (Chapter 3), we had chosen 5 initial parts with equal frequencies per variable while the criterion evolution in Figure 5.1 shows that a 5 parts based partition is less probable than a 3, 4 or 6 parts based partition, as shown by the values of the criterion before optimization. As a results, the variables neither need to be partitioned into equal number of parts nor to be partitioned with equal frequencies. For example, the variable *SepalWidth* only needs to be partitioned into two parts $SepalWidth]-\infty;3.15]$ and $SepalWidth]3.15;+\infty[$, with 95 and 55 observations, respectively.

2. **Optimized co-clustering structure.** The structure resulting from the first approach contains 3 clusters of instances and 8 clusters of variable parts while the optimized clustering structure, which is more probable, contains 3 clusters of instances and only 7 clusters of variable parts. Furthermore, the counts per co-cluster are better tuned. The optimized model (criterion value 7459.8) is more probable than the initial model (7462.6) and more probable than the model starting with 5 parts per variable (7551.8). As a result, the mutual information in the optimized model provides a better association between the clusters of instances and their most representative clusters of variable parts.

3. **Detection of the origins of the variable parts.** Whenever two (or more) parts of a variable are contained within the same cluster of variable parts, they are merged if they belong to a categorical variable or if they are contiguous otherwise, which is the case here for the variable parts $SepalWidth]-\infty;2.85]$ and $SepalWidth]2.85;3.15]$ which are merged to create $SepalWidth]-\infty;3.15]$.

4. **More precise conclusions about the data.** Without a measure for evaluating the partitioning, it would be impossible to guess the right number of parts per variable or the right boundaries. While the two approaches describe three clusters of instances containing small, medium, and large Iris flowers, the boundaries of the variable parts describing the clusters in both cases are not the same. For example, from the first approach the small flowers are described by the variable parts $Class\{setosa\}$, $PetalLength]-\infty;1.55]$ and $PetalWidth]-\infty;0.25]$, while from the optimized model, the cluster of small flowers is explained by the parts $Class\{setosa\}$, $PetalLength]-\infty;2.4]$, and $PetalWidth]-\infty;0.8]$.

5. **More probable clustering of the instances.** The clustering resulting from the first approach (50, 54, and 46 instances per cluster) is not as precise as the optimized one (50, 51, and 49 instances per cluster). This is mainly due to the fact that the partitioning is optimized. Similarly, the number of observations associated to each co-cluster is more precise (compare Figure 5.2 with Figure 5.3).

The Iris data set is small and the patterns it contains are relatively simple to recover, which makes the distinction between the two models rather subtle, although visible. In section 5.2.2, with the Adult data set, the distinction between the two models will be more easily visible because the data is complex and relatively large.

Comparison with CrossCat

For comparison, we applied CrossCat on the Iris data set. To apply CrossCat, two main user-parameters need to be specified, namely the number of Markov chains and the number of transitions. The number of chains defines

the number of samples returned by CrossCat from the posterior model distribution. We set the number of transitions to 500 and the number of chains to 20. These values fall within the range recommended by the authors (10-100 independent samples and 100-1000 transitions per chain) and provide a trade-off between stability of the results and the computation time.

On the Iris data set, CrossCat provides a single view for the data (1 cluster of variables) which correctly retrieves the interdependence between the variables. However, it produces 7 to 13 homogeneous clusters of instances as shown in Figure 5.4. Furthermore, the recovered clusters are of largely variable sizes as shown in Figure 5.5. Clusters containing one single instance are in fact very common. This is not a limitation of the CrossCat model. It simply follows from the different natures of the two models.

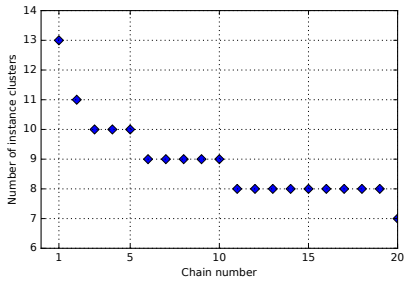


Figure 5.4 – *Iris: number of clusters of instances (CrossCat).*

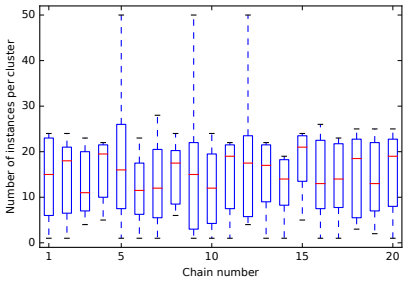


Figure 5.5 – *Iris: distribution of the number of instances per cluster (CrossCat).*

Comparison. To summarize, CrossCat detects the correlation between all the variables in the Iris data set. However, it provides a very detailed clustering of instances, for each of the 20 posterior samples. Our proposed MAP co-clustering model provides a better trade-off between the number of produced clusters of instances (and of variable parts) on one hand and expressing the intrinsic structure of the data on the other hand. Indeed, our optimized model provides just as many variable parts and as many clusters (of instances as well as variable parts) as necessary to summarize the data set. For example, Figure 5.6 shows the inferred optimum partitions of the most informative variables (*PetalLength*, *PetalWidth*, and *Class*) with respect to our co-clustering. Figure 5.7 shows an example of projected CrossCat clustering on the same variables, for one of the 20 samples. The projection of the data set, along with our optimized clustering of instances, emphasizes the correspondence between the recovered clusters (shown by color) compared to the *Class* values (shown by different markers) and shows the existing mixture between *virginica* and *versicolor*. The projected CrossCat clustering shows a global homogeneity of the clusters but a large variance in the cluster sizes (the diamond shaped black cluster contains 50 instances while the circled pink cluster contains 1 instance).

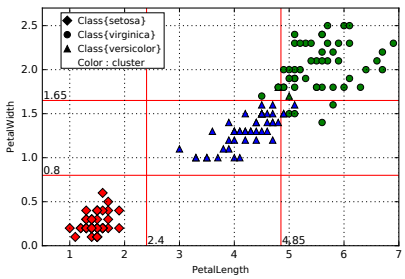


Figure 5.6 – *Our model: projection of the Iris flowers.*

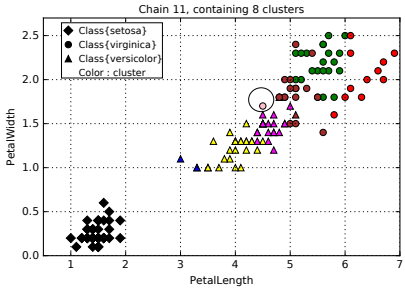


Figure 5.7 – *CrossCat: projection of the Iris flowers.*

The difference in results is natural since our approach uses a MAP approach to find the best fitting parameters while CrossCat is a full Bayes model for density estimation. In short, our model focuses on scalability, interpretability and robustness while the full Bayes CrossCat model could be more useful in other aspects such as joint density estimation and missing value imputation (see Mansinghka et al. (2016)).

Next, we report on the experimental results obtained with our co-clustering model on medium and large sized data sets.

5.2.2 Adult data

In this section, we discuss the results obtained on the relatively large data set Adult. The Adult data set (Lichman (2013)) is composed of $I = 48.842$ instances represented by $K_n = 6$ numerical variables and $K_c = 9$ categorical ones. The data set is extracted from the 1994 Current Population Surveys conducted by the U.S. Census Bureau (Kohavi 1996), and it is available in the UCI Machine Learning Repository (Lichman 2013). See Section 3.5.2 for a detailed description of this data set.

Initialization

Following our proposed optimization strategy, we started by partitioning variable domains into a predefined set of partition sizes ranging from 2 parts to 128 parts per variable. In particular, we used the set $\mathbf{J} = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 16, 32, 64, 128\}$. Among these initial models, the best model in terms of criterion corresponds to the solution starting with 64 parts per variable (Figure 5.8).

The optimal co-clustering

Before optimization, the initial co-clustering (resulting from the MODL co-clustering) contained 265 variable parts, 61 clusters of instances, and 73 clusters of variable parts (i.e 61×73 co-clusters). Starting from this initial solution, we follow the optimization strategy described in Section 4.2.7.

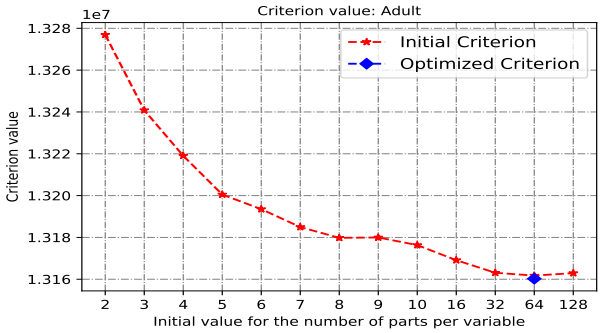


Figure 5.8 – *Adult: evolution of the criterion values.*

After optimization, the obtained optimal co-clustering contains 61×72 co-clusters and only 107 variable parts. This relatively low number of parts facilitates the process of cluster interpretation. However, with 61 clusters of instances and 72 clusters of variable parts, the co-clustering is still very detailed. Figure 5.9 shows this co-clustering in terms of mutual information.

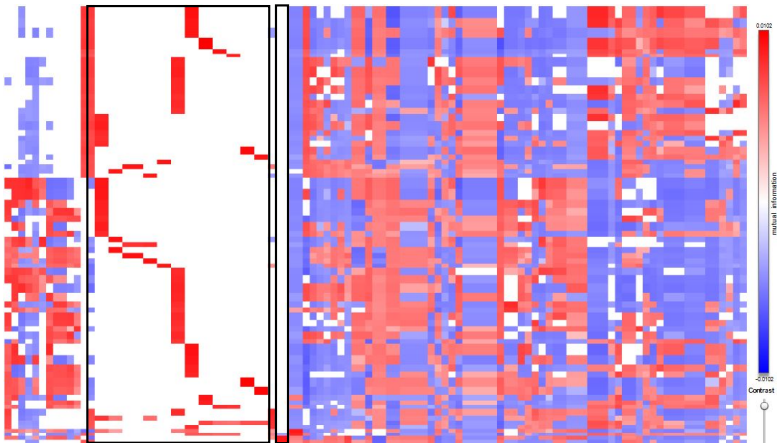


Figure 5.9 – *Adult: the optimal co-clustering. Each square represents a co-cluster. This optimal co-clustering contains* 61×72 *co-clusters.*

At this level of detailed co-clustering we recover the strong correlation between the variables *education* and *education_num* which is expressed by the existence of 13 clusters of variable parts that contain a simultaneous partitioning of these two variables. These co-clusters are shown by the highlighted zones in Figure 5.9. Their composition and the corresponding contingency table are given by Table 5.5 and Table 5.6.

This first insight demonstrates the ability of the model to detect strong dependencies between the variables and to cluster variable of different types in a meaningful way. The two variables are partitioned accordingly and their respective values are consistently grouped together (see Table 5.5). Notice, from the counts per co-cluster (Table 5.6), that these two variables create almost diagonal co-clusters.

Cluster	Variable parts
C^p_{13}	$education\_num]9.5;10.5]$, $education\{Some-college\}$
C^p_{14}	$education\{11th\}$, $education\_num]6.5;7.5]$
C^p_{15}	$education\_num]4.5;5.5]$, $education\_num]7.5;8.5]$, $education\{12th+9th\}$
C^p_{16}	$education\{10th\}$, $education\_num]5.5;6.5]$
C^p_{17}	$education\_num]3.5;4.5]$, $education\{7th-8th\}$
C^p_{18}	$education\{HS-grad\}$, $education\_num]8.5;9.5]$
C^p_{19}	$education\_num]12.5;13.5]$, $education\{Bachelors\}$
C^p_{20}	$education\{Masters\}$, $education\_num]13.5;14.5]$
C^p_{21}	$education\{Prof-school\}$, $education\_num]14.5;15.5]$
C^p_{22}	$education\{Doctorate\}$, $education\_num]15.5;+\infty[$
C^p_{23}	$education\_num]10.5;11.5]$, $education\{Assoc-voc\}$
C^p_{24}	$education\_num]11.5;12.5]$, $education\{Assoc-acdm\}$
C^p_{26}	$education\_num]-\infty;3.5]$, $education\{Preschool+1st-4th+5th-6th\}$

Table 5.5 – *Adult: composition of the variable part clusters, showing the strong interdependence between the variables* education_num *and* education. *The categorical parts delimited by a plus sign (+) are the result of multiple parts that are merged in the optimization step into one.*

Model coarsening: a simplified co-clustering

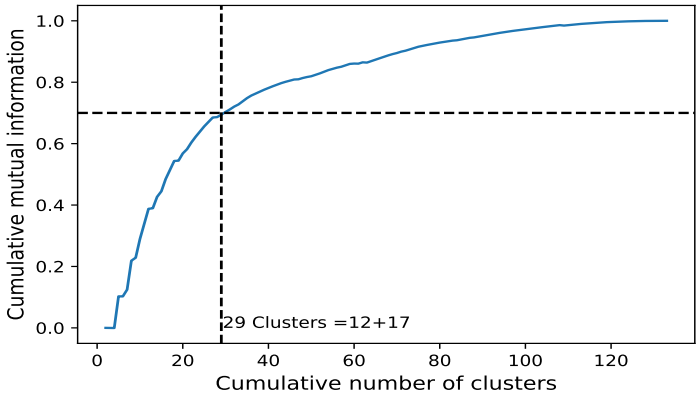


Figure 5.10 – *Adult: cumulative mutual information per co-clustering structure.*

With 61 clusters of instances and 72 clusters of variable parts, the co-clustering remains too detailed. To facilitate the analysis, the coarsening approach outlined in Section 3.3.1 is applied progressively. We proceed the analysis using two levels of granularities. The first provides a good compromise between interpretability and details: a co-clustering (Figure 5.11) that contains 12 clusters of instances, 17 clusters of variable parts, and captures 70% of information as shown in Figure 5.10. The second level of granularity is achieved starting from the first one by merging clusters of instances only to create 2 clusters of instances and keep the 17 clusters of variable parts as shown in Figure 5.12.

The associated counts of number of observations per co-cluster are given in Table 5.8 and Table 5.9. The composition of the clusters of instances is given in Table 5.7. The composition of the clusters of variable parts is given in Appendix A.

C^p_{13}	C^p_{14}	C^p_{15}	C^p_{16}	C^p_{17}	C^p_{18}	C^p_{19}	C^p_{20}	C^p_{21}	C^p_{22}	C^p_{23}	C^p_{24}	C^p_{26}
0	0	0	0	0	0	2900	0	0	0	0	0	0
0	0	0	0	0	0	1904	0	0	0	0	0	0
0	0	0	0	0	0	2284	0	0	0	0	0	0
0	0	0	0	0	0	0	2684	0	0	0	0	0
0	0	0	0	0	0	0	0	1192	0	0	0	0
0	0	0	0	0	0	0	0	0	808	0	0	0
0	0	0	0	0	2714	0	0	0	0	0	0	0
0	0	0	0	0	2134	0	0	0	0	0	0	0
0	0	0	0	0	1276	0	0	0	0	0	0	0
0	0	0	0	0	2702	0	0	0	0	0	0	0
0	0	0	0	0	2584	0	0	0	0	0	0	0
0	0	0	0	0	1244	0	0	0	0	0	0	0
0	0	0	0	0	1982	0	0	0	0	0	0	0
0	0	0	0	0	1840	0	0	0	0	0	0	0
0	0	0	0	0	1390	0	0	0	0	0	0	0
0	0	0	0	0	1722	0	0	0	0	0	0	0
0	0	0	0	0	1324	0	0	0	0	0	0	0
0	0	0	0	0	1312	0	0	0	0	0	0	0
0	0	0	0	0	2210	0	0	0	0	0	0	0
0	0	0	0	0	2122	0	0	0	0	0	0	0
0	0	0	0	0	1896	0	0	0	0	0	0	0
0	0	0	0	0	1238	0	0	0	0	0	0	0
1686	0	0	0	0	0	0	0	0	0	0	0	0
1524	0	0	0	0	0	0	0	0	0	0	0	0
2158	0	0	0	0	0	0	0	0	0	0	0	0
1918	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	896	366	0	0	0	0	0
120	0	0	0	0	342	462	0	2	0	0	0	0
0	0	0	0	0	1536	0	0	0	0	0	0	0
972	0	0	0	0	0	0	0	0	0	0	0	0
0	88	122	90	0	0	0	0	0	0	258	226	0
0	0	0	0	0	0	0	0	0	0	1772	0	0
0	0	0	0	0	0	0	0	0	0	0	1170	0
0	0	0	0	1006	0	0	0	0	0	0	0	0
0	984	0	0	0	0	0	0	0	0	0	0	0
0	0	974	0	0	0	0	0	0	0	0	0	0
0	0	0	954	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2666	0	0	0	0	0	0
0	0	0	0	0	0	2636	0	0	0	0	0	0
0	0	0	0	0	0	2302	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	2092	0	0
0	0	0	0	0	0	0	0	0	0	0	1806	0
0	0	0	0	0	0	0	1278	0	0	0	0	0
0	0	0	0	0	0	0	984	0	0	0	0	0
0	0	0	0	0	0	0	0	474	380	0	0	0
2242	0	0	0	0	0	0	0	0	0	0	0	0
2330	0	0	0	0	0	0	0	0	0	0	0	0
1240	0	0	0	0	0	0	0	0	0	0	0	0
1344	0	0	0	0	0	0	0	0	0	0	0	0
1756	0	0	0	0	0	0	0	0	0	0	0	0
1398	0	0	0	0	0	0	0	0	0	0	0	0
1714	0	0	0	0	0	0	0	0	0	0	0	0
1354	0	0	0	0	0	0	0	0	0	0	0	0
0	1200	0	0	0	0	0	0	0	0	0	0	0
0	0	490	542	0	0	0	0	0	0	0	0	0
0	1352	0	0	0	0	0	0	0	0	0	0	0
0	0	1240	0	0	0	0	0	0	0	0	0	0
0	0	0	1192	0	0	0	0	0	0	0	0	0
0	0	0	0	904	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	0	0	0	0	934
0	0	0	0	0	0	0	0	0	0	0	0	744

Table 5.6 – *Adult: counts per co-cluster, showing the strong dependence between the variables* education_num *and* education.

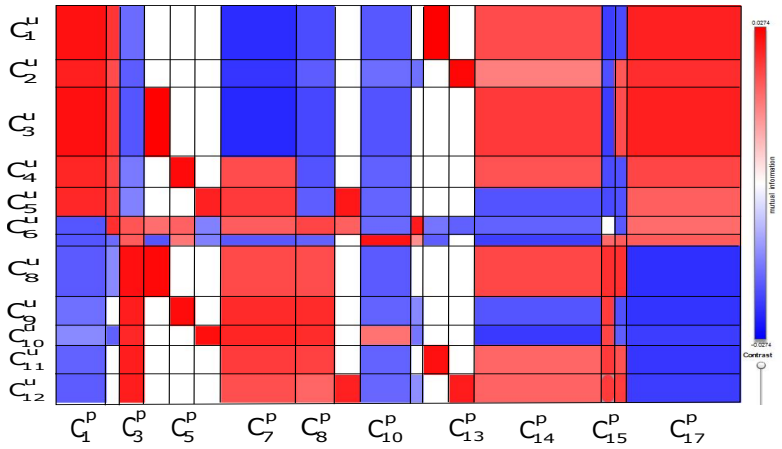


Figure 5.11 – *A co-clustering of the Adult data set containing* 12×17 *co-clusters.*

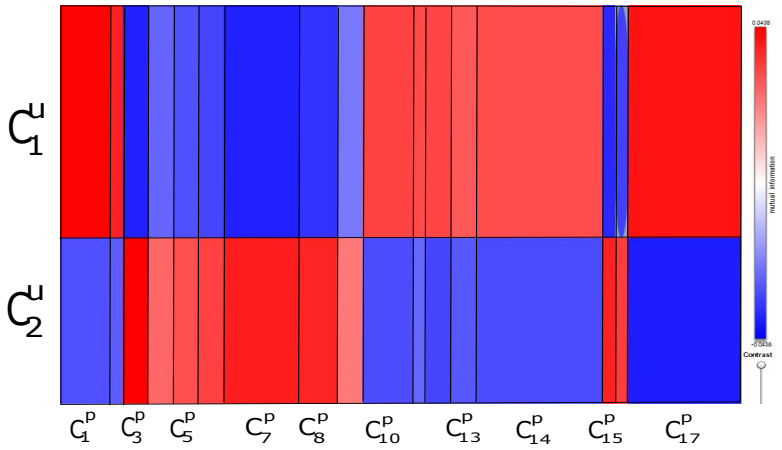


Figure 5.12 – *A co-clustering of the Adult data set containing* 2×17 *co-clusters.*

Cluster $C^u_{g_u}$	C^u_1	C^u_2	C^u_3	C^u_4	C^u_5	C^u_6	C^u_7	C^u_8	C^u_9	C^u_{10}	C^u_{11}	C^u_{12}
$\lvert C^u_{g_u} \rvert$	6689	3460	8518	3802	3507	2277	1303	6327	3544	2342	3643	3430

Cluster $C^u_{g_u}$	C^u_1	C^u_2
$\lvert C^u_{g_u} \rvert$	28253	20589

Table 5.7 – *Adult: composition of the clusters of instances in the* 12×17 *co-clustering (top) and the* 2×17 *co-clustering (bottom).*

Cluster	C_1^p	C_2^p	C_3^p	C_4^p	C_5^p	C_6^p	C_7^p	C_8^p	C_9^p	C_{10}^p	C_{11}^p	C_{12}^p	C_{13}^p	C_{14}^p	C_{15}^p	C_{16}^p	C_{17}^p	*marginal*
C_1^u	14936	3686	13	0	0	0	7059	318	0	375	0	13378	0	41790	3003	1291	14486	100335
C_2^u	7881	1627	51	0	0	0	2846	76	0	323	30	0	6920	21321	1833	1333	7659	51900
C_3^u	17723	4274	60	17036	0	0	9027	381	0	311	0	0	0	53992	4244	3170	17552	127770
C_4^u	7516	1954	4	0	7604	0	6514	708	0	197	0	0	0	23815	1848	269	6601	57030
C_5^u	7033	1961	2	0	0	3116	6817	852	3898	237	0	0	0	21217	1546	401	5525	52605
C_6^u	87	2277	2277	1536	896	366	3755	1285	484	125	2167	972	300	13866	0	215	3547	34155
C_7^u	1122	348	1324	342	462	4	1769	207	0	3127	67	120	0	6888	955	600	2210	19545
C_8^u	47	1	12638	12654	0	0	10559	2389	0	148	0	0	0	39697	6326	3819	6627	94905
C_9^u	9	0	7064	0	7088	0	8384	3036	0	80	2	0	0	21499	3544	413	2041	53160
C_{10}^u	1	64	4616	0	0	4684	6882	2401	0	316	64	0	0	12786	2278	73	965	35130
C_{11}^u	30	0	7265	0	0	0	6961	1943	0	95	0	7286	0	22556	3643	1472	3394	54645
C_{12}^u	49	0	6818	0	0	0	5812	1153	2942	264	1	0	3918	21266	3430	1995	3802	51450
marginal	56434	16192	42132	31568	16050	8170	76385	14749	7324	5598	2331	21756	11138	300693	32650	15051	74409	732630

Table 5.8 – *Adult: number of observations per co-cluster in the* 12×17 *co-clustering.*

	C_1^p	C_2^p	C_3^p	C_4^p	C_5^p	C_6^p	C_7^p	C_8^p	C_9^p	C_{10}^p	C_{11}^p	C_{12}^p	C_{13}^p	C_{14}^p	C_{15}^p	C_{16}^p	C_{17}^p	*marginal*
C_1^u	56298	16127	3731	18914	8962	3486	37787	3827	4382	4695	2264	14470	7220	182889	13429	7279	57580	443340
C_2^u	136	65	38401	12654	7088	4684	38598	10922	2942	903	67	7286	3918	117804	19221	7772	16829	289290
marginal	56434	16192	42132	31568	16050	8170	76385	14749	7324	5598	2331	21756	11138	300693	32650	15051	74409	732630

Table 5.9 – *Adult: number of observations per co-cluster in the* 2×17 *co-clustering.*

Analysis of the results

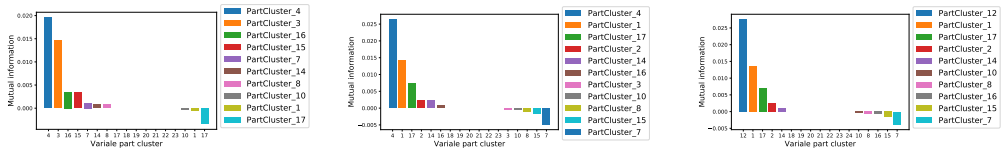
For simplicity, we start by explaining the co-clustering with 2 clusters of instances (Figure 5.12) which provides a high level view of the data, then we zoom in on the 12×17 co-clustering for more detailed conclusions.

The 2×17 co-clustering: from Figure 5.12, the clusters of variable parts C_1^p, C_2^p, C_3^p, C_8^p, C_{15}^p, and C_{17}^p oppose two major clusters of instances. For example, the cluster of variable parts C_2^p contains the variable part $sex\{Female\}$ and it is over represented in the cluster of instances C_1^u and under represented in the cluster of instances C_2^u. Inversely, the cluster of variable parts C_{15}^p contains the variable part $sex\{Male\}$ which is under represented in the cluster of instances C_1^u and over represented in the cluster C_2^u. These two clusters of variable parts oppose the female individuals from males. More precisely, we distinguish:

- a cluster C_1^u of 28253 instances containing individuals who are characterized by C_1^u, C_2^p, and C_{17}^p. This cluster can be described as: unmarried (C_1^p) females (C_2^p) who are likely to gain less than $50K$ a year (C_{17}^p),
- a cluster C_2^u of 20589 instances containing individuals who are characterized by C_3^p, C_8^p, and C_{15}^p. This cluster can be described as: married (C_3^p) males (C_{15}^p) who are likely to gain more than $50K$ a year (C_8^p).

The 12×17 co-clustering: for interpretation, one can view the co-clustering from the perspective of the instances, or from the perspective of the variable parts. Here, we provide examples of the conclusions that can be retrieved from both dimensions separately.

The clusters of instances. For simplicity, only the most informative clusters of instances are explained here. A detailed description of the 12 clusters can be found in Appendix A.



(a) C_8^u (b) C_3^u (c) C_1^u

Figure 5.13 – *Adult: examples of ranking clusters of instances by clusters of variable parts.*

The most informative clusters of instances are C_8^u, C_3^u, and C_1^u, which can be characterized by their most contributing clusters of variable parts (Figure 5.13) as follows:

- C_8^u is characterized by C_4^p and C_3^p. Thus, it contains individuals who are high school graduates, have about 9 years of education, and are married males.

  1. C_4^p contains the variable parts: $education\{HS-grad\}$, $education\_num]8.5;9.5]$.
  2. C_3^p contains the variable parts: $marital\_status\{Married-civ-spouse + Married-AF-spouse\}$, $relationship\{Husband\}$.

- C_3^u is characterized by C_4^p and C_1^p. Thus, it contains young individuals who are high school graduates, have around 9 years of education, and are not married.

  1. C_4^p contains the variable parts detailed above.
  2. C_1^p contains the variable parts: $age]-\infty;21.5]$, $relationship\{Own-child + Unmarried + Not-in-family\}$, $marital\_status\{Separated + Married-spouse-absent + Divorced + Widowed + Never-married\}$.

- C_1^u is characterized by C_{12}^p and C_1^p. Thus, it contains young individuals who have some college degree, have around 10 years of education, and are not married.

  1. C_{12}^p contains the variable parts: $education\_num]9.5;10.5]$, $education\{Some-college\}$.
  2. C_1^p is as described above.

The clusters of variable parts. As seen earlier with the 2×17 co-clustering, the clusters of variable parts C_1^p, C_2^p, C_3^p, C_8^p, C_{15}^p, and C_{17}^p oppose the clusters of instances C_1^u to C_6^u from the clusters C_7^u to C_{12}^u.

- The cluster of variable parts C_2^p contains the part $sex\{Female\}$ and it is over represented in the clusters of instances C_1^u to C_6^u and under represented in the clusters of instances C_7^u to C_{12}^u. The clusters of instances C_1^u to C_6^u can thus be characterized as *Females*. Note that inversely, the cluster of variable parts C_{15}^p contains the variable part $sex\{Male\}$ and is under represented in the clusters of instances C_1^u to C_6^u and over represented in the clusters C_7^u to C_{12}^u.

- The clusters of variable parts C_1^p and C_3^p oppose unmarried females in the clusters C_1^u to C_5^u and married males (husbands) in the clusters C_7^u to C_{12}^u. The cluster of instances C_6^u is a special case since it contains married females characterized by the variable part cluster C_{11}^p which contains the part $relationship\{Wife\}$.

- The clusters of variable parts C_8^p and C_{17}^p oppose the clusters of instances who are likely to gain less than $50K$ a year (clusters C_1^u to C_5^u) and those who are likely to gain more (clusters C_6^u to C_{12}^u).

These oppositions can be seen clearly from the simplified co-clustering (Figure 5.12).

The most correlated variables. As seen from the optimal co-clustering, the simplified co-clustering still shows a correlation between the variables *education* and *education_num*. The clusters C_4^p, C_5^p, C_6^p , C_9^p , C_{12}^p, and C_{13}^p express this correlation. Furthermore, one can notice that each of these variable parts is present in a set of male instances and a set of female instances. Table 5.10 shows the composition of the co-clusters expressing this correlation.

Cluster	C_4^p	C_5^p	C_6^p	C_9^p	C_{12}^p	C_{13}^p
C_1^u	0	0	0	0	13378	0
C_2^u	0	0	0	0	0	6920
C_3^u	17036	0	0	0	0	0
C_4^u	0	7604	0	0	0	0
C_5^u	0	0	3116	3898	0	0
C_6^u	1536	896	366	484	972	300
C_7^u	342	462	4	0	120	0
C_8^u	12654	0	0	0	0	0
C_9^u	0	7088	0	0	0	0
C_{10}^u	0	0	4684	0	0	0
C_{11}^u	0	0	0	0	7286	0
C_{12}^u	0	0	0	2942	0	3918

Table 5.10 – *Adult: content of the co-clusters expressing the correlation between* education *and* education_num *from the simplified* 12×17 *co-clustering.*

Summary. In summary, note that these are only brief examples of the insights one can gain from the model. In fact, with a maximum of 61 clusters of instances, 72 clusters of variable parts, and with the many levels of the hierarchy of instance and variable parts clusters, lies a wide range of co-clustering models for the exploratory analyst to study. Thus, a wide range of explainable clusters and co-clusters.

From the analysis we performed on the 12×17 co-clustering, we have detected strong dependence between variables of different types, a non information variable grouped in one part (the cluster C_{14}^p contains the variable part *fnlwgt*$]-\infty;+\infty]$), and easily interpretable clusterings of the instances.

Comparison with the co-clustering methodology

In comparison with the approach proposed in Chapter 3, we notice that globally, the interpretations of the clusters are similar. However, below are some of the aspects in which the two approaches differ.

1. **The initial number of parts.** In the formalized model, the number of variable parts and the partition of the variables is optimized automatically as a model parameter. In the first approach (Chapter 3), we

had chosen 10 initial parts with equal frequencies per variable. The criterion evolution in Figure 5.1 shows that a 10 parts based partition is less probable than a 16, 32, 64 or 128 parts based partition, as explained by the value of the criterion.

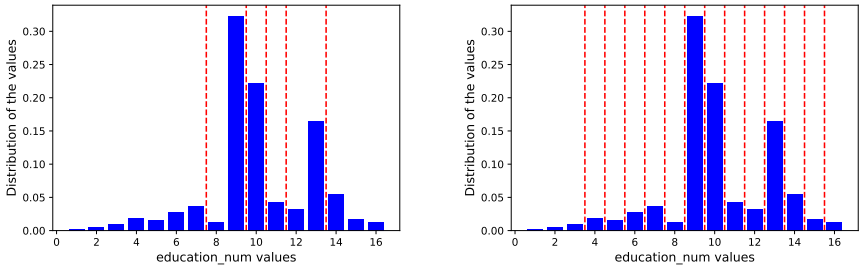
2. **The optimized number of parts.** The variables do not need to be partitioned into equal number of parts or with equal frequencies per part. Tables 5.11 and 5.12 show the number of parts per variable and examples of the number of observations per part which shows that the optimized partitioning is not equal frequency. Also, Figure 5.14 shows a comparison between the optimized partition and the parameter based one. The full list of the number of observations per part is given in Appendix A.

variable	*capital_gain*	*hours_per_week*	*race*	*age*	*capital_loss*	*fnlwgt*	*relationship*
number of parts	5	10	5	10	3	3	6
marital_status	*native_country*	*education*	*sex*	*workclass*	*class*	*education_num*	*occupation*
5	10	13	2	6	2	14	13

Table 5.11 – *Adult: the optimized number of parts per variable.*

Part	Observations
$age]-\infty;18.5]$	1457
⋮	⋮
$age]32.5;39.5]$	9073
⋮	⋮
$age]63.5;+\infty]$	2427
$class\{less\}$	37155
$class\{more\}$	11687
$hours\_per\_week]-\infty;29.5]$	6151
⋮	⋮
$hours\_per\_week]39.5;40.5]$	22803
⋮	⋮
$hours\_per\_week]59.5;+\infty]$	3853

Table 5.12 – *Adult: examples of the number of observations per part.*



(a) *Partitioning with 10 parts.* (b) *Optimized partitioning.*

Figure 5.14 – *Adult: comparison of the optimized partition of the variable* education_num *with the parameter based one.*

3. **The co-clustering structure is optimized.** The structure resulting from the first approach contained 34×62 co-clusters while the optimized model contains 61×72 co-clusters. Also the counts of number of observations per co-cluster differ greatly. The former co-clustering (with criterion value 13176327.26) is less probable than the latter (with criterion value 13160261.72).

4. **Detection of the origins of the variable parts.** In the optimization phase, whenever two (or more) parts of a variable are contained within the same cluster of variable parts, they are merged if they belong to a categorical variable or if they are contiguous otherwise. For example, in the initial model the variable *age* contained 47 parts while in the optimized model, it contains only 10 parts. Overall, the initial model contained 265 parts while the optimized one contains 107 parts.

5. **More tuned bounds of the parts.** Without a measure for evaluating the partitioning, it would be impossible to guess the right number of parts per variable or the right boundaries. When explaining the clusters of instances, the optimized partitions provide more precise description of the clusters since their bounds are better tuned.

Next, we report on the experimental results obtained with our co-clustering model on the large data set CensusIncome. Unfortunately, we could not run the CrossCat model on the Adult data set (nor on CensusIncome) for scalability reasons. The process stops after hours of computation. Therefore, this section presents only the results of our co-clustering model.

5.2.3 CensusIncome data

To evaluate the scalability of the co-clustering model on a large data set, we now apply it to the CensusIncome data set. The CensusIncome data set is a larger version of the Adult data studied in the previous section. It contains census data extracted from the 1994 and 1995 Current Population Surveys conducted by the U.S. Census Bureau (Kohavi 1996). The data is available in the UCI Machine Learning Repository (Lichman 2013) and it is composed of $I = 299.285$ instances represented by 8 numerical and 34 categorical demographic and employment related variables. The data includes 624.096 missing values, thus a total of $N = 11.945.874$ observations is considered. As for the Adult data set, this data set is frequently used to test supervised classification methods where the goal is to predict whether a person makes *more* or *less* than $50K$ a year (the variable *class*). We build the co-clustering with the *class* variable but not in a supervised way. To our model, the data simply contains 42 variables of equal importance. As a consequence, the class information can be used to validate the obtained results.

The co-clustering results

Since the data set is considerably large, the first optimization step is applied with the predefined number of parts from 2 to 128, by power of 2. The optimal model is found at the starting point corresponding to a maximum of 64 parts per variable. The optimized model contains 249 variable parts, 607 clusters of instances and 97 clusters of variable parts. Figure 5.15 shows the color coded version of the mutual information. Extracting relevant information from this many clusters and co-clusters is a tedious task. Thus, we simplify the model to a reasonable 12×12 co-clusters (Figure 5.16).

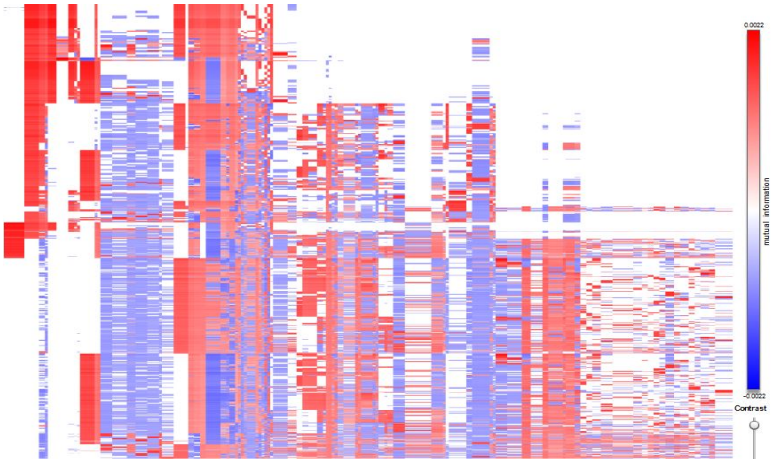


Figure 5.15 – *CensusIncome: the optimized co-clustering. Each square represents a co-cluster. This optimal co-clustering contains* 607×97 *co-clusters.*

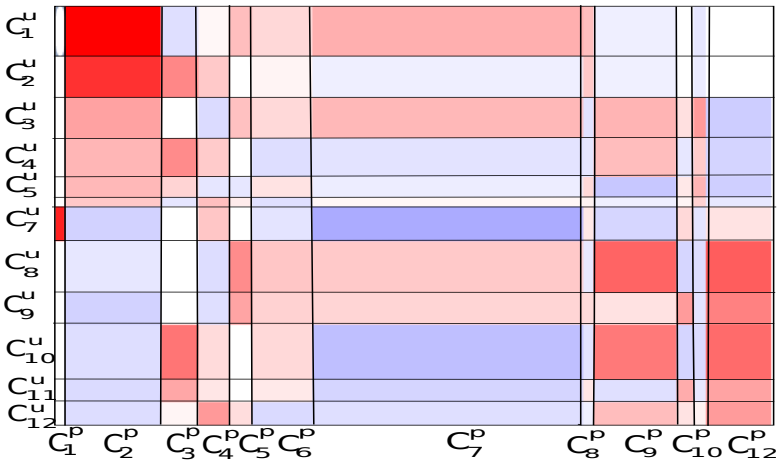


Figure 5.16 – *CensusIncome:* 12×12 *co-clusters.*

Analysis of the results

To analyze the data set, we use the simplified model containing 12 clusters of instances and 12 clusters of variable parts (Figure 5.16). An example of easily interpretable results from Figure 5.16, can be derived from the

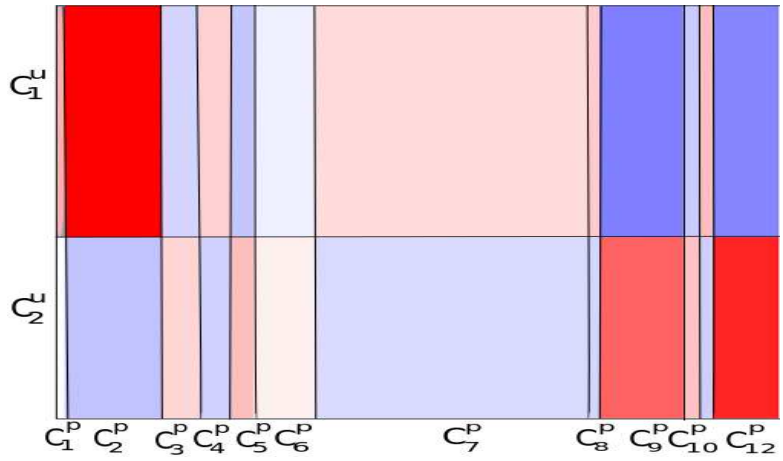


Figure 5.17 – *CensusIncome:* 2×12 *co-clusters.*

fact that the 2^{nd} cluster of variable parts C_2^p is over represented on the clusters of instances from C_1^u to C_6^u and under represented on the clusters of instances from C_7^u to C_{12}^u. On the other hand, the 12^{th} cluster of variable parts (C_{12}^p) is under represented on the instance clusters from C_1^u to C_6^u and over represented on the clusters from C_7^u to C_{12}^u. This indicates that these clusters of variable parts can be used to distinguish two major groups of instances, as shown in Figure 5.17, if one merged all the corresponding clusters of instances. The clusters of instances C_1^u and C_{10}^u are the most representative of these major clusters (as they contain the most represented co-clusters in Figure 5.16). Each of these two clusters of instances can be explained by the most represented clusters of variable parts.

The cluster C_1^u contains individuals who, based on the co-clusters with the highest contribution to the mutual information, can be characterized as follows:

- from (C_1^u, C_2^p): less than 15 years old, not in university, do not work and are not tax payers,
- from (C_1^u, C_7^p): males and females with low capital gain (less than 57), low capital loss (less than 70), low wage per hour (less than 10), who earn less than $50K$ ($Class\{-50000\}$), are not enrolled in a university, do not have their own business, and are not self employed.

The cluster of instances C_{10}^u contains individuals who, based on the co-clusters with the highest contribution to the mutual information, can be characterized as follows:

- from (C_{10}^u, C_3^p): have stayed in the same house for over a year (called non mover),
- from (C_{10}^u, C_4^p) and (C_{10}^u, C_6^p): are united states citizens by naturalization or native born,

- from (C_{10}^u, C_9^p): aged from 27 to 64, have high education (from high school graduate to doctorate degree), are married, pay taxes as a couple, earn more than $50K$ a year, have a high capital loss and a high capital gain,
- from (C_{10}^u, C_{12}^p): work more than 30 weeks a year, with a wage of more than 10 dollars per hour, and work in the local, state or federal government.

Overall, this first insight clearly separates the active and inactive individuals. Note that the CensusIncome data is considerably large. Within an optimal model of 607 clusters of instances and 97 clusters of variable parts, lies a wide range of exploitable insights, depending on the level of details one chooses to consider in their exploratory analysis. We do not provide a full recount of the conclusions that can be retrieved from this data set because our objective is not to study this data set but rather to show that the co-clustering model applies to large data sets and still provides interpretable results.

5.3 Experiments on artificial data sets: comparison with CrossCat

In this section, we apply the co-clustering model to artificial data sets to evaluate its behavior in extreme cases. Namely, in the case of data that contains no clusters and in the case of perfectly correlated variables. The results are compared with those obtained with the CrossCat model. The objective here is to show the robustness of the co-clustering model and its capacity to extract the correlations between variables. We show that the co-clustering model consistently detects an absence of pattern when the variables are independent, by creating one co-cluster and one part per variable, and that it approximates the correlated variables with increasing numbers of clusters.

5.3.1 The data

In order to evaluate the behavior of our co-clustering model, with respect to structure extraction, experiments have been conducted on artificial data. To this end, synthetic data is generated according to different scenarios reflecting different types of patterns. The first scenario contains only independent variables. Each categorical variable is sampled uniformly from a set of possible values V_p (say $|V_p| = p + 1$, where p is the index of the variable) whereas each numerical variable is sampled from a continuous uniform distribution on the interval $[0; 1]$. The data generated according to this scenario contains no particular pattern.

The second scenario contains strongly correlated variables. We start by sampling a numerical variable X_1 (according to a uniform distribution

$[0; 1]$). The remaining variables are obtained from X_1 as follows:

$$\begin{cases} X_p = X_1 & \text{for } 2 \leq p \leq K_n = \frac{P}{2}, \\ Y_p = \lfloor (p+1)X_1 \rfloor & \text{for } 1 \leq p \leq K_c = \frac{P}{2}, \end{cases}$$

where P is the number of variables, X_p denotes a numerical variable, Y_p denotes a categorical variable, and $\lfloor x \rfloor$ denotes the integer part of the numerical value x.

This ensures a strong correlation between the numerical variable and the categorical values. Given this configuration, we expect the retrieved co-clusters to be diagonal and the number of co-clusters to increase with the number of observations in the data.

For both scenarios, samples are generated with a varying number of variables P (from 2 to 32 by power of 2) and varying number of independent instances I (from 16 to 4096, by power of two).

Our co-clustering model is applied with an initial number of variable parts ranging from 2 to 512, by power of 2. As mentioned earlier, for CrossCat, we set the number of sampled models to 20 and the number of transitions to 500 per sample.

5.3.2 Results

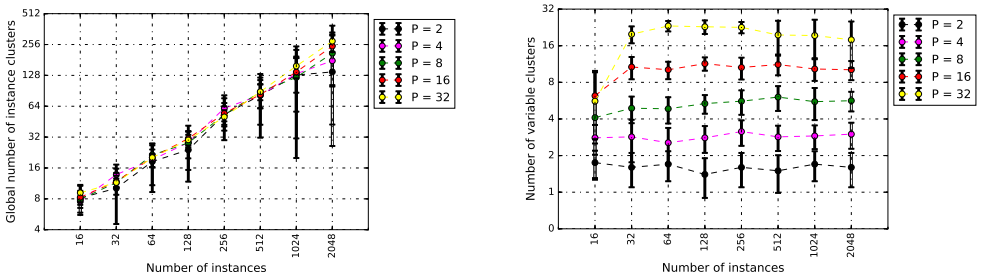
We measure the ability of a model to extract the data structure by the number of different produced clusters.

To evaluate our resulting models, we collect the number of clusters of instances, the number of clusters of variable parts, and consequently the number of co-clusters as a function of the number of variables and the number of instances in the data. For CrossCat, we collect the number of views (clusters of variables) in every model and the number of clusters of instances per view. The mean and standard deviation of each collected measure, over the 20 samples obtained in a run, are considered for a final measure of the number of instance/variable clusters.

Independent variables

In the case of independent variables, our co-clustering model systematically detects the absence of structure by producing one cluster of instances, one cluster of variable parts, and one part per variable.

Using CrossCat, the number of variable clusters (views) increases as the number of variables increases which is expected since the variables are independent. However, total independence between the variables is not detected as the number of clusters is inferior to the number of variables. The number of clusters of instances increases also with the number of instances in the data. For example, it produces more than one hundred clusters of instances for a data set containing 1000 instances (Figure 5.18).



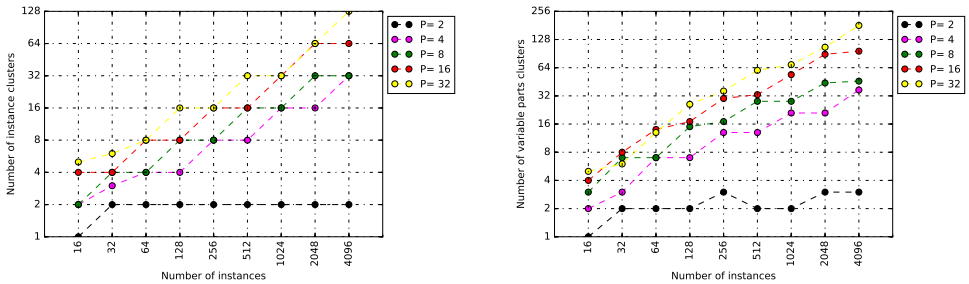
(a) *Mean and standard deviation of the number of instance clusters*

(b) *Mean and standard deviation of the number of variable clusters*

Figure 5.18 – *CrossCat on independent variables.*

Correlated variables

For the strongly correlated data, the number of recovered clusters of instances and of variables parts, retrieved by our co-clustering model, increases as expected with the number of instances and with the number of variables as shown in Figure 5.19.



(a) *Number of instance clusters*

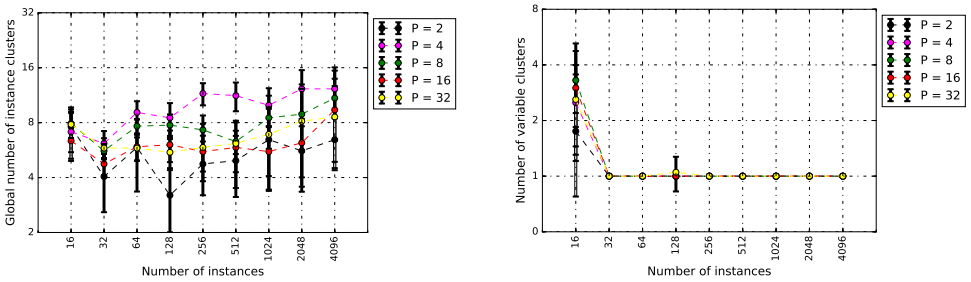
(b) *Number of variable part clusters*

Figure 5.19 – *Our co-clustering on correlated variables.*

For very small data sets (16 instances), CrossCat detects 1 to 6 views in the data which corresponds to 1 to 6 independent sets of mutually correlated variables. Given larger data, CrossCat correctly detects the correlation with one view containing all variables. However, the number of instance clusters is in average less than 10 (with large variance), whatever the number of instances and of variables. More unexpectedly, the accuracy of the retrieved correlation structure does not improve with the number of instances or variables (Figure 5.20).

5.3.3 Comparison of the results

The experiments conducted on artificial data confirms the expected behavior of our co-clustering model in the extreme cases of pattern-less data and



(a) *Mean and standard deviation of the number of instance clusters*

(b) *Mean and standard deviation of the number of variable clusters*

Figure 5.20 – *CrossCat on correlated variables: the number of clusters of instances and variables.*

perfectly correlated variables. When the data contains no structure, our co-clustering model systematically returns a single co-cluster. On perfectly correlated variables, the number of recovered clusters of instances and clusters of variable parts increases systematically with the number of observations in the data. In comparison, CrossCat detects the correlation between the variables. However, it produces too many clusters of instances when there is no pattern and produces fewer clusters than expected when the variables are strongly correlated.

Contrasting our model with CrossCat

It is worth noting that our experiments focus on interpretability of the results and robustness rather than other aspects of data analysis, such as missing value imputation and density estimation. Thus, further experiments are required for a full comparison between the proposed co-clustering model and CrossCat. The difference in the experimental results follows from the difference in nature of the two models.

- CrossCat is a full Bayes model while our proposed model uses a maximum a posteriori estimation.
- CrossCat is a multi-vue approach while our model provides a single summarizing view of the data.
- CrossCat provides a clustering at the level of variables and multiple clusterings at the level of instances. Our model provides a clustering at the level of variable parts and one clustering at the level of instances. Because the outer clustering of CrossCat operates at the variable level, it does not allow to identify partial cross-dependencies between variables. Our clustering of the partitions of variables enable extracting such partial dependencies.
- CrossCat uses different distributions and priors than the ones we use. For example, numerical variables are modeled via Gaussian distribu-

tions in CrossCat while our model uses a much more flexible rank based non parametric approach. Also, CrossCat uses Dirichlet processes as priors for its clusterings while we use a flatter prior that allows for more balanced clustering structures (in the small data regime where priors strongly influence the results).

- While the complexity of the two models are very comparable in theory, the implementation provided by the authors (Mansinghka (2017)) could not run on large data sets (namely Adult and CensusIncome).
- CrossCat is more flexible and provides predictions for missing values but the clusterings are difficult to interpret while our model provides more interpretable results and facilitates the exploratory analysis of the data.
- The experiments conducted here do not evaluate the density estimation aspect of our model versus CrossCat. Further experiments are required to compare these aspects.

In summary, while closely related, CrossCat and the proposed co-clustering model are very different.

5.4 Conclusion

In this chapter, we have applied the co-clustering model proposed in Chapter 4 to various data sets in order to highlight its main features. We have shown that the co-clustering model can discover complex dependencies between variables of different types, and provides clustering of the instances that is easily interpretable via the variable part clusters and the over/under representation diagrams. We have also shown that the model scales by applying it to data sets containing up to around 12 millions of observations. Furthermore, the analysis shows the utility of the exploration tools such as model coarsening which enables us to simplify the structure and analyze the data on various levels of granularity.

Two important aspects could be studied in further works. Namely, the choice of the initial number of parts and improvement of the optimization algorithm, which could allow to reach different solutions.

Chapter 6

Conclusion

Exploratory analysis is the first go-to approach for understanding large and complex data sets. The importance of this approach comes from the fact that, beforehand, we have no prior knowledge about the data. Afterwards, the conclusions made about the data, as a result of the exploratory analysis, can be used as they are in decision making or used as a basis for further confirmatory analysis. For example, in market research, exploratory analysis driven conclusions can provide enough insights to answer client, product or service related specific questions and to suggest (or not) launching a new product or service. Additionally, the post-exploratory understanding of the data can indicate the appropriate techniques to use for further analysis. In this thesis, we have investigated the use of co-clustering as an exploratory analysis technique.

6.1 Contribution of this thesis

In Chapter 2, we have shown that *co-clustering* methods tend to differ in their underlying assumptions and the types and shapes of clusters and co-clusters they aim to extract. In particular, some techniques aim for extracting one single co-cluster defined by a group of instances and a groups of variables that express a pattern of interest. Other techniques aim for simultaneously extracting multiple co-clusters but the overall co-clustering structure differs from one method to another.

In Chapter 3, we have proposed a co-clustering methodology that consists in homogenizing the data to create categorical variable parts, via discretization of numerical variables and value grouping for the categorical ones, then using the MODL approach to create clusters of instances and clusters of variable parts. We have shown that this approach extracts easily interpretable clusters since each cluster of instances can be associated to one or many clusters of variable parts and inversely. We have also shown that the approach can extract local and global dependencies between the variables if the homogenizing process is chosen wisely. Furthermore, the conclusions derived from this co-clustering approach are consistent with those obtained when applying a singular value decomposition based data transformation followed

by a standard k-means algorithm. However, this approach requires specifying the dicretization. Given a user defined number of parts per variable, it uses equal frequency dicretization. Hence, the accuracy of the results is dependent on this granularity parameter.

In Chapter 4, we have proposed a parameter-less co-clustering model that formalizes the approach proposed in Chapter 3. The proposed model relies on a specific data representation. Namely, it sets the entries of the data matrix (observations) as statistical units. For parameter estimation, it optimizes a MAP based model selection criterion to infer the number of parts per variable automatically, to create an optimized partitioning of the variables, and to perform a co-clustering. The model itself consists of a mapping of the instances to clusters of instances, a mapping of the variables to variable parts (ranges of values), a mapping of the variable parts to clusters of variable parts, and a mapping of the observations to the co-clusters formed at the intersection of the clusters of instances and the clusters of variable parts. As a result of the used data representation, this co-clustering enables us to handle numeric and categorical data simultaneously, and to count for missing values and set valued variables.

Chapter 5 presents experimental results on real-world data sets with increasing sizes and complexities and on artificially generated data to highlight the features of the co-clustering model in extreme cases, namely in the case of absence of pattern and in the case of perfectly correlated variables. This chapter also provides a comparison with the CrossCat model (Mansinghka et al. 2016) which relates to our model in some aspects and differs in others.

6.2 Perspectives for future work

We have shown that the proposed co-clustering model enables to extract a particular structure that is previously unseen in the literature. The extracted structure consists of a partition of the instances and a partition of partitions of variables. The evaluation criterion computes the exact probability of fitting a given set of model parameters to the data. However, it is computationally infeasible to evaluate all the possible partitions. Therefore, we have proposed a greedy iterative optimization procedure that enables us to choose the best set of parameters starting from a set of initial solutions. Hence, improvement over the optimization algorithm and/or the choice of the initial solutions could allow to reach different, potentially better, solutions. Furthermore, it would be useful to propose more elaborated exploratory analysis techniques and visualization tools for easier analysis of the co-clustering results. For example: tools for choosing the clusters to analyze and for ranking the instances within a cluster of instances (to identify the most typical instance for a cluster similarly to identifying the center in k-means); tools for ranking the variable parts within a cluster of variable parts, and associating the clusters of instances with single variable parts instead of a cluster of parts; etc.

A second promising future work is to use the co-clustering model to

generate new data. It could be interesting in privacy preserving applications to publish the generated data instead of the original data. The goal would be to ensure that no object from the original data set is directly identifiable from the released data while ensuring that data mining techniques can still be applied to the synthetic data. This approach has been used for example by Fessant et al. (2017).

Appendices

Appendix A

Adult: details of the co-clustering results

A.1 Results from the optimal co-clustering

Table A.1 shows the optimized variable partitions in the optimal co-clustering with 61×72 co-clusters. This table illustrates that the variables are neither partitioned with equal parts nor with equal frequencies.

variable X_k	J_k	P_k	$n^p_{j_k}$
capital_gain	5	*capital_gain*$]-\infty; 57]$	44807
		capital_gain$]57; 1839]$	219
		capital_gain$]1839.0; 5119.0]$	1484
		capital_gain$]5119; 7560]$	664
		capital_gain$]7560.0; +\infty]$	1668
hours_per_week	10	*hours_per_week*$]-\infty; 29.5]$	6151
		hours_per_week$]29.5; 35.5]$	4181
		hours_per_week$]35.5; 39.5]$	1355
		hours_per_week$]39.5; 40.5]$	22803
		hours_per_week$]40.5; 44.5]$	934
		hours_per_week$]44.5; 45.5]$	2717
		hours_per_week$]45.5; 49.5]$	1020
		hours_per_week$]49.5; 55.5]$	5623
		hours_per_week$]55.5; 59.5]$	205
		hours_per_week$]59.5; +\infty]$	3853
race	5	*race*$\{Other\}$	406
		race$\{Amer-Indian-Eskimo\}$	470
		race$\{Asian-Pac-Islander\}$	1519
		race$\{Black\}$	4685
		race$\{White\}$	41762
age	10	*age*$]-\infty; 18.5]$	1457
		age$]18.5; 21.5]$	3262
		age$]21.5; 23.5]$	2507
		age$]23.5; 27.5]$	4786
		age$]27.5; 32.5]$	6359
		age$]32.5; 39.5]$	9073
		age$]39.5; 46.5]$	7951
		age$]46.5; 58.5]$	8869
		age$]58.5; 63.5]$	2151
		age$]63.5; +\infty]$	2427
capital_loss	3	*capital_loss*$]-\infty; 1748.0]$	47332
		capital_loss$]1748; 1975.5]$	730
		capital_loss$]1975.5; +\infty[$	780
fnlwgt	3	*fnlwgt*$]-\infty; 61655.0]$	4578
		fnlwgt$]61655.0; 208067.5]$	27475
		fnlwgt$]208067.5; +\infty]$	16789

relationship	6	$relationship\{Other-relative\}$	1506
		$relationship\{Wife\}$	2331
		$relationship\{Unmarried\}$	5125
		$relationship\{Own-child\}$	7581
		$relationship\{Not-in-family\}$	12583
		$relationship\{Husband\}$	19716
marital_status	5	$marital\_status\{Widowed\}$	1518
		$marital\_status\{Separated\}+\{Married-spouse-absent\}$	2158
		$marital\_status\{Divorced\}$	6633
		$marital\_status\{Never-married\}$	16117
		$marital\_status\{Married-civ-spouse\}$ $+\{Married-AF-spouse\}$	22416
native_country	10	$native\_country\{France\}$	38
		$native\_country\{Ireland\}+\{Poland\}$	124
		$native\_country\{Hungary\}+\{Iran\}+\{Greece\}+\{Yugoslavia\}$	151
		$native\_country\{Trinadad\&Tobago\}$ $+\{Outlying-US\}+\{Jamaica\}+\{Haiti\}$	231
		$native\_country\{Italy\}+\{Nicaragua\}+\{Ecuador\}+\{Cuba\}$	337
		$native\_country\{Puerto-Rico\}+\{Honduras\}$ $+\{Scotland\}+\{Peru\}+\{Columbia\}$	356
		$native\_country\{Canada\}+\{Germany\}+\{England\}$	515
		$native\_country\{Vietnam\}+\{India\}+\{China\}$ $+\{Hong\}+\{Thailand\}+\{South\}+\{Taiwan\}$ $+\{Philippines\}+\{Laos\}+\{Japan\}+\{Cambodia\}$	1037
		$native\_country\{Portugal\}+\{Dominican-Republic\}$ $+\{Mexico\}+\{Guatemala\}+\{El-Salvador\}$	1364
		$native\_country\{United-States\}$	44689
education	13	$education\{Doctorate\}$	594
		$education\{Prof-school\}$	834
		$education\{Preschool\}+\{1st-4th\}+\{5th-6th\}$	839
		$education\{7th-8th\}$	955
		$education\{10th\}$	1389
		$education\{12th\}+\{9th\}$	1413
		$education\{Assoc-acdm\}$	1601
		$education\{11th\}$	1812
		$education\{Assoc-voc\}$	2061
		$education\{Masters\}$	2657
		$education\{Bachelors\}$	8025
		$education\{Some-college\}$	10878
		$education\{HS-grad\}$	15784
sex	2	$sex\{Female\}$	16192
		$sex\{Male\}$	32650
workclass	6	$workclass\{Without-pay\}+\{Never-worked\}$	31
		$workclass\{Self-emp-inc\}$	1695
		$workclass\{State-gov\}$	1981
		$workclass\{Self-emp-not-inc\}$	3862
		$workclass\{Local-gov\}+\{Federal-gov\}$	4568
		$workclass\{Private\}$	36705
class	2	$class\{more\}$	11687
		$class\{less\}$	37155
education_num	14	$education\_num]-\infty;3.5]$	839
		$education\_num]3.5;4.5]$	955
		$education\_num]4.5;5.5]$	756
		$education\_num]5.5;6.5]$	1389
		$education\_num]6.5;7.5]$	1812
		$education\_num]7.5;8.5]$	657
		$education\_num]8.5;9.5]$	15784
		$education\_num]9.5;10.5]$	10878
		$education\_num]10.5;11.5]$	2061
		$education\_num]11.5;12.5]$	1601
		$education\_num]12.5;13.5]$	8025
		$education\_num]13.5;14.5]$	2657
		$education\_num]14.5;15.5]$	834
		$education\_num]15.5;+\infty[$	594

occupation	13	$occupation\{Priv-house-serv\}$	242
		$occupation\{Protective-serv\}+\{Armed-Forces\}$	998
		$occupation\{Tech-support\}$	1446
		$occupation\{Farming-fishing\}$	1490
		$occupation\{Handlers-cleaners\}$	2072
		$occupation\{Transport-moving\}$	2355
		$occupation\{Machine-op-inspct\}$	3022
		$occupation\{Other-service\}$	4923
		$occupation\{Sales\}$	5504
		$occupation\{Adm-clerical\}$	5611
		$occupation\{Exec-managerial\}$	6086
		$occupation\{Craft-repair\}$	6112
		$occupation\{Prof-specialty\}$	8981

Table A.1 – *Adult: partitioning of the variables in the optimal co-clustering.*

A.2 Results from the simplified co-clustering

Table A.2 and Table A.3 show the mutual information expressed by the 12×17 co-clustering and the 2×17 co-clustering, respectively. Table A.4 and Table A.5 show the partitioning of the variables and the composition of the clusters of variable parts in the 12×17 co-clustering. The composition of the clusters of instances is given in Chapter 5 (Section 5.2.2, Table 5.7).

A.2.1 Interpretation of the clusters of instances

As mentioned in Section 5.2.2, we provide here a full description of the 12 clusters of instances in the simplified 12×17 co-clustering. Recall that this co-clustering captures 70% of the information included in the Adult data set. This co-clustering (from Figure 5.9, Table A.2 and Table A.5) enables us to distinguish:

1. a cluster of instances C_1^u containing 6689 individuals who are females (C_2^p), less than 23 years old, are not married, with a child (C_1^p), have around 9 years of education in Some-college (C_{12}^p), work less than 35 hours a week and gain less than $50K$ a year (C_{17}^p);

2. a cluster of instances C_2^u containing 3460 individuals who are females (C_2^p), less than 27 years old (C_1^p and C_{17}^p), are not married, with a child (C_1^p), have more than three years and less than nine years of education (C_{13}^p), are likely to work in *Farming-fishing*, *Handlers-cleaners*, *Transport-moving*, *Machine-op-inspct*, *Craft-repair* (C_{16}^p), and are likely to gain less than $50K$ a year (C_{17}^p);

3. a cluster of instances C_3^u containing 8518 individuals who are females (C_2^p), less than 33 years old (C_1^p, C_{14}^p and C_{17}^p), are not married, with a child (C_1^p), have around 9 years of education and some high school degree (*HS-grad*) (C_4^p), are likely to work in *Farming-fishing*, *Handlers-cleaners*, *Transport-moving*, *Machine-op-inspct*, *Craft-repair* (C_{16}^p), and to gain less than $50K$ a year (C_{17}^p);

Cluster	C_1^p	C_2^p	C_3^p	C_4^p	C_5^p	C_6^p	C_7^p	C_8^p	C_9^p	C_{10}^p	C_{11}^p	C_{12}^p	C_{13}^p	C_{14}^p	C_{15}^p	C_{16}^p	C_{17}^p	*marginal*
C_1^u	0.013	0.003	0	0	0	0	-0.004	-0.001	0	-0.0	0	0.028	0	0.001	-0.002	-0.001	0.007	0.044
C_2^u	0.007	0.001	-0.0	0	0	0	-0.002	-0.0	0	-0.0	0	0	0.02	0.0	-0.001	0.0	0.004	0.029
C_3^u	0.014	0.002	-0.0	0.026	0	0	-0.005	-0.001	0	-0.0	0	0	0	0.002	-0.002	0.001	0.007	0.044
C_4^u	0.006	0.001	0	0	0.019	0	0.001	-0.0	0	-0.0	0	0	0	0.001	-0.001	-0.001	0.001	0.027
C_5^u	0.005	0.001	0	0	0	0.007	0.002	-0.0	0.011	-0.0	0	0	0	-0.0	-0.001	-0.001	0.0	0.024
C_6^u	-0.0	0.003	0.0	0.0	0.0	-0.0	0.0	0.001	0.0	-0.0	0.009	-0.0	-0.0	-0.0	0	-0.0	0.0	0.013
C_7^u	-0.0	-0.0	0.0	-0.0	0.0	0	-0.0	-0.0	0	0.013	0.0	-0.0	0	-0.001	0.0	0.0	0.0	0.012
C_8^u	-0.0	0	0.015	0.02	0	0	0.001	0.001	0	-0.0	0	0	0	0.001	0.003	0.003	-0.003	0.041
C_9^u	0	0	0.008	0	0.018	0	0.005	0.004	0	-0.0	0	0	0	-0.0	0.002	-0.001	-0.003	0.033
C_{10}^u	0	-0.0	0.005	0	0	0.016	0.006	0.004	0	0.0	-0.0	0	0	-0.002	0.001	-0.0	-0.002	0.028
C_{11}^u	0	0	0.008	0	0	0	0.002	0.002	0	-0.0	0	0.015	0	0.0	0.002	0.001	-0.002	0.028
C_{12}^u	-0.0	0	0.008	0	0	0	0.001	0.0	0.007	-0.0	0	0	0.008	0.0	0.002	0.002	-0.002	0.026
marginal	0.045	0.011	0.044	0.046	0.037	0.023	0.007	0.01	0.018	0.013	0.009	0.043	0.028	0.002	0.003	0.003	0.007	0.349

Table A.2 – *Adult: mutual information of the* 12×17 *co-clustering.*

	C_1^p	C_2^p	C_3^p	C_4^p	C_5^p	C_6^p	C_7^p	C_8^p	C_9^p	C_{10}^p	C_{11}^p	C_{12}^p	C_{13}^p	C_{14}^p	C_{15}^p	C_{16}^p	C_{17}^p	*marginal*
C_1^u	0.038	0.011	-0.01	-0.0	-0.001	-0.002	-0.01	-0.004	-0.0	0.002	0.001	0.002	0.001	0.001	-0.007	-0.002	0.019	0.039
C_2^u	-0.001	-0.0	0.044	0.0	0.001	0.002	0.013	0.009	0.0	-0.001	-0.0	-0.002	-0.001	-0.001	0.01	0.003	-0.013	0.063
marginal	0.037	0.011	0.034	0.0	0.0	0.0	0.003	0.005	0.0	0.001	0.001	0.0	0.0	0.0	0.003	0.001	0.006	0.102

Table A.3 – *Adult: mutual information of the* 2×17 *co-clustering.*

4. a cluster of instances C_4^u containing 3802 individuals who are females (C_2^p) of all ages (C_1^p, C_7^p, C_{14}^p and C_{17}^p), not married, with a child (C_1^p), have bachelor's degree (C_5^p), are as likely to work around 40 hours a week (C_{14}^p) and to gain less than $50K$ a year (C_{17}^p);

5. a cluster of instances C_5^u containing 3507 individuals who are females (C_2^p) of all ages (C_1^p, C_7^p, C_{14}^p and C_{17}^p), are not married, with a child (C_2^p), have conducted more than 10 years of studies (C_6^p and C_9^p), have *Doctorate*, *Masters* or *Prof-school* degree, and are likely to work as *Exec-managerial*, *Prof-specialty*, *Protective-serv*, *Armed-Forces* or *Tech-support*, more than 40 hours a week (C_7^p), and are likely to gain less than 50K a year (C_{17}^p);

6. a cluster of instances C_6^u containing 2277 individuals who are females (C_2^p), are married (C_{11}^p), are more than 32 years old (C_7^p), and are likely to work as *Exec-managerial*, *Prof-specialty*, *Protective-serv*, *Armed-Forces* or *Tech-support* (C_7^p), and to gain more (C_8^p) or less than $50K$ a year (C_{17}^p);

7. a cluster of instances C_7^u containing 1303 individuals who have no education (less than 4 years and up to the 6^{th} grade) (C_{10}^p), and gain less than $50K$ a year (C_{17}^p);

8. a cluster of instances C_8^u containing 6327 individuals who are probably married (C_3^p), have conducted around 8 years of studies and have a *HS-grad* grade (C_4^p), males (C_{15}^p), and are likely to work as *Exec-managerial*, *Prof-specialty*, *Protective-serv*, *Armed-Forces* or *Tech-support* (C_7^p) or to work in *Farming-fishing* , *Handlers-cleaners*, *Transport-moving*, *Machine-op-inspct*, *Craft-repair* (C_{16}^p). These individuals work more than 40 hours a week (C_7^p), and are likely to gain more than $50K$a year (C_8^p);

9. a cluster of instances C_9^u containing 3544 individuals who have high probability of being: married (C_3^p), males (C_{15}^p), with bachelor's degree (C_4^p), gain more than $50K$ a year (C_8^p), are more than 32 years old (C_7^p), work more than 40 hours per week as *Exec-managerial*, *Prof-specialty*, *Protective-serv*, *Armed-Forces* or *Tech-support* (C_8^p);

10. a cluster of instances C_{10}^u containing 2342 individuals who have high probability of being married (C_3^p), have conducted long studies (more than 13 years), have *Masters*, *Doctorate* or *Prof-school* degree (C_6^p), males (C_{15}^p), gain more than $50K$ a year (C_8^p), are more than 32 years old, work more than 40 hours per week as *Exec-managerial*, *Prof-specialty*, *Protective-serv*, *Armed-Forces* or *Tech-support* (C_7^p);

11. a cluster of instances C_{11}^u containing 3643 individuals who are probably married (C_3^p), have conducted around 9 years of studies in *Some-college* (C_{12}^p), males (C_{15}^p), and are likely to work in *Farming-fishing* , *Handlers-cleaners*, *Transport-moving*, *Machine-op-inspct*, *Craft-repair*

(C_{16}^p) or as *Exec-managerial*, *Prof-specialty*, *Protective-serv*, *Armed-Forces* or *Tech-support* (C_7^p), work more than 40 hours a week (C_7^p), and are likely to gain more than $50K$ a year (C_8^p);

12. a cluster of instances C_{12}^u containing 3430 individuals who are probably married (C_3^p), have more than three years and less than ten years of education (C_9^p and C_{13}^p), males (C_{15}^p), are likely to work in *Farming-fishing*, *Handlers-cleaners*, *Transport-moving*, *Machine-op-inspct*, *Craft-repair* (C_{16}^p) or as *Exec-managerial*, *Prof-specialty*, *Protective-serv*, *Armed-Forces* and *Tech-support* (C_7^p), more than 40 hours a week (C_7^p), and to gain more than $50K$ a year(C_8^p).

variable X_k	J_k	P_k	n^p_{Jk}
capital_gain	4	*capital_gain*$]-\infty;57]$	44807
		capital_gain$]57;1839]$	219
		capital_gain$]1839.0;5119.0]$	1484
		capital_gain$]5119.0;+\infty]$	2332
hours_per_week	3	*hours_per_week*$]-\infty;39.5]$	11687
		hours_per_week$]39.5;40.5]$	22803
		hours_per_week$]40.5;+\infty]$	14352
race	3	$race\{Asian-Pac-Islander\}$	1519
		$race\{Black\}+\{Amer-Indian-Eskimo\}$	5155
		$race\{Other\}+\{White\}$	42168
age	4	*age*$]-\infty;21.5]$	4719
		age$]21.5;27.5]$	7293
		age$]27.5;32.5]$	6359
		age$]32.5;+\infty]$	30471
capital_loss	3	*capital_loss*$]-\infty;1748.0]$	47332
		capital_loss$]1748;1975.5]$	730
		capital_loss$]1975.5;+\infty[$	780
fnlwgt	1	*fnlwgt*$]-\infty;+\infty]$	48842
relationship	4	$relationship\{Other-relative\}$	1506
		$relationship\{Wife\}$	2331
		$relationship\{Husband\}$	19716
		$relationship\{Own-child\}+\{Unmarried\}$ $+\{Not-in-family\}$	25289
marital_status	2	$marital\_status\{Married-civ-spouse\}$ $+\{Married-AF-spouse\}$	22416
		$marital\_status\{Separated\}+\{Married-spouse-absent\}$ $+\{Divorced\}+\{Widowed\}+\{Never-married\}$	26426
native_country	4	$native\_country\{Trinadad\&Tobago\}+\{Outlying-US\}$ $+\{Jamaica\}+\{Haiti\}+\{Puerto-Rico\}+\{Honduras\}$ $+\{Scotland\}+\{Peru\}+\{Columbia\}$	587
		$native\_country\{Hungary\}+\{Iran\}+\{Greece\}+\{Yugoslavia\}$ $+\{Ireland\}+\{Poland\}+\{France\}+\{Italy\}+\{Nicaragua\}$ $+\{Ecuador\}+\{Cuba\}+\{Canada\}+\{Germany\}+\{England\}$	1165
		$native\_country\{Vietnam\}+\{India\}+\{China\}+\{Hong\}$ $+\{Thailand\}+\{South\}+\{Taiwan\}+\{Philippines\}$ $+\{Laos\}+\{Japan\}+\{Cambodia\}+\{Portugal\}+\{Mexico\}$ $+\{Dominican-Republic\}+\{Guatemala\}+\{El-Salvador\}$	2401
		$native\_country\{United-States\}$	44689
education	7	$education\{Preschool\}+\{1st-4th\}+\{5th-6th\}$	839
		$education\{Assoc-acdm\}+\{Assoc-voc\}$	3662
		$education\{Prof-school\}+\{Masters\}+\{Doctorate\}$	4085
		$education\{7th-8th\}+\{10th\}+\{11th\}+\{12th\}+\{9th\}$	5569
		$education\{Bachelors\}$	8025
		$education\{Some-college\}$	10878
		$education\{HS-grad\}$	15784
sex	2	$sex\{Female\}$	16192
		$sex\{Male\}$	32650
workclass	3	$workclass\{Without-pay\}+\{Never-worked\}$	31
		$workclass\{State-gov\}+\{Self-emp-inc\}$ $+\{Self-emp-not-inc\}+\{Local-gov\}$ $+\{Federal-gov\}$	12106
		$workclass\{Private\}$	36705
class	2	$class\{more\}$	11687
		$class\{less\}$	37155
education_num	7	*education_num*$]-\infty;3.5]$	839
		education_num$]3.5;8.5]$	5569
		education_num$]8.5;9.5]$	15784
		education_num$]9.5;10.5]$	10878
		education_num$]10.5;12.5]$	3662
		education_num$]12.5;13.5]$	8025
		education_num$]13.5;+\infty]$	4085
occupation	4	$occupation\{Sales\}$	5504
		$occupation\{Priv-house-serv\}+\{Other-service\}$ $+\{Adm-clerical\}$	10776
		$occupation\{Machine-op-inspct\}+\{Craft-repair\}$ $+\{Transport-moving\}+\{Handlers-cleaners\}$ $+\{Farming-fishing\}$	15051
		$occupation\{Tech-support\}+\{Protective-serv\}$ $+\{Armed-Forces\}+\{Prof-specialty\}+\{Exec-managerial\}$	17511

Table A.4 – *Adult: partitioning of the variables in the* 12×17 *co-clustering.*

Cluster	Variable parts
C_1^p	$age]-\infty;21.5]$, $relationship\{Own-child\}+\{Unmarried\}$ $+\{Not-in-family\}$, $marital\_status\{Separated\}+\{Married-spouse-absent\}$ $+\{Divorced\}+\{Widowed\}+\{Never-married\}$
C_2^p	$sex\{Female\}$
C_3^p	$marital\_status\{Married-civ-spouse\}+\{Married-AF-spouse\}$, $relationship\{Husband\}$
C_4^p	$education\{HS-grad\}$, $education\_num]8.5;9.5]$
C_5^p	$education\_num]12.5;13.5]$, $education\{Bachelors\}$
C_6^p	$education\{Prof-school\}+\{Masters\}+\{Doctorate\}$, $education\_num]13.5;+\infty]$
C_7^p	$occupation\{Tech-support\}+\{Protective-serv\}+\{Armed-Forces\}$ $+\{Prof-specialty\}+\{Exec-managerial\}$, $workclass\{State-gov\}$ $+\{Self-emp-inc\}+\{Self-emp-not-inc\}+\{Local-gov\}$ $+\{Federal-gov\}$, $hours\_per\_week]40.5;+\infty]$, $native\_country\{Hungary\}$ $+\{Iran\}+\{Greece\}+\{Yugoslavia\}+\{Ireland\}+\{Poland\}+\{France\}$ $+\{Italy\}+\{Nicaragua\}+\{Ecuador\}+\{Cuba\}+\{Canada\}+\{Germany\}$ $+\{England\}$, $age]32.5;+\infty]$, $capital\_loss]1975.5;+\infty[$
C_8^p	$class\{more\}$, $capital\_loss]1748;1975.5]$, $capital\_gain]5119.0;+\infty]$
C_9^p	$education\_num]10.5;12.5]$, $education\{Assoc-acdm+Assoc-voc\}$
C_{10}^p	$education\_num]-\infty;3.5]$, $education\{Preschool\}+\{1st-4th\}+\{5th-6th\}$, $race\{Asian-Pac-Islander\}$, $native\_country\{Vietnam\}+\{India\}+\{China\}$ $+\{Hong\}+\{Thailand\}+\{South\}+\{Taiwan\}+\{Philippines\}+\{Laos\}$ $+\{Japan\}+\{Cambodia\}+\{Portugal\}+\{Dominican-Republic\}+\{Mexico\}$ $+\{Guatemala\}+\{El-Salvador\}$
C_{11}^p	$relationship\{Wife\}$
C_{12}^p	$education\_num]9.5;10.5]$, $education\{Some-college\}$
C_{13}^p	$education\_num]3.5;8.5]$, $education\{7th-8th\}+\{10th\}+\{11th\}+\{12th\}+\{9th\}$
C_{14}^p	$age]27.5;32.5]$, $capital\_loss]-\infty;1748.0]$, $native\_country\{United-States\}$, $capital\_gain]-\infty;57]$, $fnlwgt]-\infty;+\infty]$, $workclass\{Private\}$, $occupation\{Sales\}$, $race\{Other\}+\{White\}$, $hours\_per\_week]39.5;40.5]$, $capital\_gain]1839.0;5119.0]$
C_{15}^p	$sex\{Male\}$
C_{16}^p	$occupation\{Machine-op-inspct\}+\{Craft-repair\}+\{Transport-moving\}$ $+\{Handlers-cleaners\}+\{Farming-fishing\}$
C_{17}^p	$relationship\{Other-relative\}$, $workclass\{Without-pay\}+\{Never-worked\}$, $class\{less\}$, $capital\_gain]57;1839]$, $occupation\{Priv-house-serv\}$ $+\{Other-service\}+\{Adm-clerical\}$, $race\{Black\}+\{Amer-Indian-Eskimo\}$, $native\_country\{Trinadad\&Tobago\}+\{Outlying-US\}+\{Jamaica\}+\{Haiti\}$ $+\{Puerto-Rico\}+\{Honduras\}+\{Scotland\}+\{Peru\}+\{Columbia\}$, $hours\_per\_week]-\infty;39.5]$, $age]21.5;27.5]$

Table A.5 – *Adult: composition of the clusters of variable parts in the* 12×17 *co-clustering.*

Appendix B

Model based co-clustering of mixed numerical and binary data

Model based co-clustering of mixed numerical and binary data

Aichetou Bouchareb, Marc Boullé, Fabrice Clérot and Fabrice Rossi

Abstract Co-clustering is a data mining technique used to extract the underlying block structure between the rows and columns of a data matrix. Many approaches have been studied and have shown their capacity to extract such structures in continuous, binary or contingency tables. However, very little work has been done to perform co-clustering on mixed type data. In this article, we extend the latent block models based co-clustering to the case of mixed data (continuous and binary variables). We then evaluate the effectiveness of the proposed approach on simulated data and we discuss its advantages and potential limits.

1 Introduction

The goal of co-clustering is to jointly perform a clustering of rows and a clustering of columns of a data table. Proposed by [Good, 1965] then by [Hartigan, 1975], co-clustering is an extension of the standard clustering that extracts the underlying structure in the data in the form of clusters of row and clusters of columns. The advantage of this technique, over the standard clustering, lies in the *joint (simultaneous)* analysis of the rows and columns which enables extracting the maximum of information about the interdependence between the two entities. The utility of co-clustering lies in its capacity to create easily interpretable clusters and its capability to reduce a large data table into a significantly smaller matrix having the same structure as the orig-

Aichetou Bouchareb, Marc Boullé and Fabrice Clérot:
Orange Labs, 2 Avenue Pierre Marzin 22300 Lannion - France, e-mail: `firstname.lastname@orange.com`

Fabrice Rossi, Aichetou Bouchareb:
SAMM EA 4534 - University of Paris 1 Panthéon-Sorbonne, 90 rue Tolbiac 75013 Paris - France, e-mail: `firstname.lastname@univ-paris1.fr`

inal data. Performing an analysis on the smaller summary matrix enables the data analyst to indirectly study the original data while significantly reducing the cost in space and computing time.

Since its introduction, many co-clustering methods have been proposed (for example, [Bock, 1979, Cheng and Church, 2000, Dhillon et al., 2003]). These methods differ mainly in the type of data (continuous, binary or contingency data), in the considered hypotheses, the method of extraction and the expected results (hard clustering, fuzzy clustering, hierarchical clustering, etc.). One of the renowned approaches is the co-clustering using latent block models which is a mixture model based technique where each cluster of rows or columns is defined by latent variables to estimate ([Govaert and Nadif, 2013]). These models extend the use of Gaussian mixture models and Bernoulli mixture models to the context of co-clustering.

Latent block based co-clustering models have therefore been proposed and validated for numerical, binary, categorical, and contingency data. Nevertheless, to our knowledge, these models have never been applied to mixed data. Actually, real life data is not always either numerical or categorical and an outright information extraction method is required to handle mixed type data as well as uni-typed data. Since the majority of data analysis methods are designed for a particular type of input data, the analyst finds himself/herself forced to go through a phase of data pre-processing to transform the data into a uni-type data (often binary) in order to use an appropriate method. Another option is to separately analyze each part of the data (by type) using an appropriate method, then perform a joint interpretation of the results. However, data pre-processing is very likely to result in a loss of information while independently analyzing different parts of the data, using methods that are based on different models, makes the joint interpretation of the results even harder and sometimes the results are simply incoherent.

Mixture models have been used to analyze mixed data in the context of clustering, by [McParland and Gormley, 2016] who propose using a latent variable model according to the Gaussian distribution regardless of the data type (numerical, binary, ordinal, or nominal data). However, the use of these models in co-clustering remains uncommon. In this paper, we propose to extend the co-clustering mixture models, proposed by [Govaert and Nadif, 2003, Govaert and Nadif, 2008], to the case of mixed data (with numerical and binary variables) by adopting the same approach of maximum likelihood estimation as the authors.

The remainder of this paper is organized as follows. In section 2, we start by defining the latent block models and their use in co-clustering. In section 3, we extend these models to mixed data co-clustering. Section 4 presents our experimental results on simulated data. Section 5 provides a discussion of the results. Finally, conclusions and future work are presented in section 6.

2 Latent block model based co-clustering

Consider the data table $\mathbf{x} = (x_{ij}, i \in I, j \in J)$ where I is the set of n objects and J the set of d variables characterizing the objects, defined by the rows and columns of the matrix $\mathbf{x}$, respectively. The goal of co-clustering is to find a partition $\mathcal{Z}$ of the rows into g groups and a partition $\mathcal{W}$ of the columns into m groups, describing the permutation of rows and columns that defines groups of rows and groups of columns and forms homogeneous *blocks* at the intersections of the groups. Supposing the number of row clusters and the number of column clusters to be known, an entry x_{ij} belongs to the block $B_{kl} = (I_k, J_l)$ if and only if the row $x_{i.}$ belongs to the group I_k of rows, and the column $x_{.j}$ belongs to the group J_l of columns. The partitions of the rows and columns can be represented by the binary matrix $\mathbf{z}$ of row affiliations to the row clusters and the binary matrix of column affiliations $\mathbf{w}$, where $z_{ik} = 1$ if and only if $x_{i.} \in I_k$ and $w_{jl} = 1$ if and only if $x_{.j} \in J_l$.

The likelihood of the latent block model LMB is given by:

$$f(\mathbf{x}; \theta) = \sum_{(\mathbf{z},\mathbf{w}) \in \mathcal{Z} \times \mathcal{W}} p((\mathbf{z}, \mathbf{w}); \theta) p(\mathbf{x} | \mathbf{z}, \mathbf{w}; \theta), \tag{1}$$

where θ is the set of unknown model parameters, and $\mathcal{Z} \times \mathcal{W}$ is the set of all possible partitions $\mathbf{z}$ of I and $\mathbf{w}$ of J that fulfill the following LBM hypotheses:

1. the existence of a partition of rows into g clusters $\{I_1, \ldots, I_g\}$ and a partition of columns into m clusters $\{J_1, \ldots, J_m\}$ such that each entry x_{ij}, of the data matrix, is the result of a probability distribution that depends only on its row cluster and its column cluster. These partitions can be represented by latent variables that can be estimated,
2. the memberships of the row clusters and of the column clusters are independent,
3. knowing the cluster memberships, the observed data units are independent (conditional independence to the couple ($\mathbf{z}$, $\mathbf{w}$)).

Under these hypotheses, the log-likelihood of the data is given by:

$$L(\theta) = \log f(\mathbf{x}; \theta) = \log \left(\sum_{(\mathbf{z},\mathbf{w}) \in \mathcal{Z} \times \mathcal{W}} \prod_{ik} \pi_k^{z_{ik}} \prod_{jl} \rho_l^{w_{jl}} \prod_{ijkl} \varphi_{kl}(x_{ij}; \alpha_{kl})^{z_{ik} w_{jl}} \right),$$

where the sums and products over i, j, k, and l have their limits from 1 to n, d, g, and m, respectively, π_k and ρ_l are the proportions of the k^{th} cluster of rows and the l^{th} cluster of columns, and α_{kl} is the set of parameters specific to the block B_{kl}. The likelihood φ_{kl} is that of a Gaussian distribution in the case of numerical data and that of a Bernoulli distribution in the case of binary data.

For an $(n \times d)$ data matrix and a partition into $g \times m$ co-clusters, the sum over $\mathcal{Z} \times \mathcal{W}$ would take at least $g^n \times m^d$ operations ([Brault and Lomet, 2015]).

Directly computing the log-likelihood is infeasible in a reasonable time, preventing therefore a direct application of EM algorithm, classically used in mixture models. Thus, [Govaert and Nadif, 2008] use a variational approximation and a Variational Expectation Maximization algorithm for optimization.

3 LBM based co-clustering of mixed data

The latent block model as defined in Section 2 can only be applied to unitype data. To extend their use to the case of mixed data, we now consider a mixed type data table $\mathbf{x} = (x_{ij}, i \in I, j \in J = J_c \cup J_d)$ where I is the set of n objects characterized by continuous and binary variable, J_c is the set of d_c continuous variables and J_d the set of d_d binary variables. Our goal is to find a partition of rows into g clusters, a partition of the continuous columns into m_c clusters, and a partition of the binary columns into m_d clusters, denoted $\mathcal{Z}$, $\mathcal{W}_c$ and $\mathcal{W}_d$ respectively.

Additionally to the previously mentioned LBM hypotheses, we suppose that the partition of rows, the partition of continuous columns and the partition of the binary columns are independent. These partitions are represented by the binary clustering matrices $\mathbf{z}$, $\mathbf{w}_c$, $\mathbf{w}_d$ and by the fuzzy clustering matrices s, tc and td, respectively. Furthermore, conditionally on $\mathbf{w}_c$, $\mathbf{w}_d$ and $\mathbf{z}$, the data matrix entries $(x_{ij})_{\{i\in I, j\in J\}}$ are supposed independent and there is a mean, independent from the model, to distinguish the continuous columns from the binary ones. Under these hypotheses, the likelihood of the generative model for mixed data can be written as:

$$f(\mathbf{x};\theta) = \sum_{(\mathbf{z},\mathbf{w}_c,\mathbf{w}_d)\in\mathcal{Z}\times(\mathcal{W}_c,\mathcal{W}_d)} \left(\prod_{ik} \pi_k^{z_{ik}} \prod_{j_c l_c} \rho_{l_c}^{wc_{j_c l_c}} \prod_{j_d l_d} \rho_{l_d}^{wd_{j_d l_d}} \right.$$
$$\left. \prod_{ij_c kl_c} \varphi^c_{kl_c}(x_{ij_c};\alpha_{kl_c})^{z_{ik} wc_{j_c l_c}} \prod_{ij_d kl_d} \varphi^d_{kl_d}(x_{ij_d};\alpha_{kl_d})^{z_{ik} wd_{j_d l_d}} \right).$$

Note that the aforementioned hypotheses lead to a simple combination of the previously existing situations (binary and continuous). Therefore, this combination adds no further mathematical difficulty, but rather potential practical consequences, resulting from coupling two different distributions (in the clustering of rows) and by the incommensurable natures of the densities (continuous variables) and probabilities (binary variables).

For likelihood optimization, we use an iterative Variational Expectation Maximization algorithm, inspired by [Govaert and Nadif, 2008], as described below.

3.1 Variational approximation

In the latent block model, the goal is to optimize the full-information, which requires knowing the latent variables $\mathbf{z}$, $\mathbf{w}_c$ and $\mathbf{w}_d$. The full-information log-likelihood is given by:

$$L_c(\mathbf{x}, \mathbf{z}, \mathbf{w}_c, \mathbf{w}_d; \theta) = \sum_{ik} z_{ik} \log \pi_k + \sum_{j_c l_c} wc_{j_c l_c} \log \rho_{l_c} + \sum_{j_d l_d} wd_{j_d l_d} \log \rho_{l_d}$$
$$+ \sum_{i j_c k l_c} z_{ik} wc_{j_c l_c} \log \varphi^c_{k l_c}(x_{i j_c}; \alpha_{k l_c}) + \sum_{i j_d k l_d} z_{ik} wd_{j_d l_d} \log \varphi^d_{k l_d}(x_{i j_d}; \alpha_{k l_d}),$$

where the sums over i, j_c, j_d, k, l_c, l_d have their limits from 1 to n, d_c, d_d, g, m_c and m_d respectively.

However, a direct application of the EM algorithm is impractical due to the dependency between the memberships to the clusters of rows $\mathbf{z}$ and the memberships to the clusters of continuous columns $\mathbf{w}_c$ on one hand, and between the memberships to the clusters of rows $\mathbf{z}$ and the memberships to the clusters of binary columns $\mathbf{w}_d$, on the other hand. This makes the computation of the joint distribution $p(\mathbf{z}, \mathbf{w}_c, \mathbf{w}_d | \mathbf{x}, \theta)$ rather an impossible task. It is thus impractical to integrate the log-likelihood of the full-information data, given this distribution.

As in [Govaert and Nadif, 2008], we use a variational approximation that consists of approximating the conditional distributions of the latent variables to a factorisable form. More precisely, we approximate $p(\mathbf{z}, \mathbf{w}_c, \mathbf{w}_d | \mathbf{x}, \theta)$ by the adjustable distribution product $q(\mathbf{z}|\mathbf{x}, \theta)$, $q(\mathbf{w}_c|\mathbf{x}, \theta)$ and $q(\mathbf{w}_d|\mathbf{x}, \theta)$, of parameters $s_{ik} = q(z_{ik} = 1|\mathbf{x}, \theta)$, $tc_{jl} = q(wc_{jl} = 1|\mathbf{x}, \theta)$ and $td_{jl} = q(wd_{jl} = 1|\mathbf{x}, \theta)$ respectively.

The full-information likelihood is thus lower bounded by the following F_c criterion

$$\begin{aligned} F_c(s, tc, td, \theta) = & \sum_{ik} s_{ik} \log \pi_k + \sum_{j_c l_c} tc_{j_c l_c} \log \rho_{l_c} + \sum_{j_d l_d} td_{j_d l_d} \log \rho_{l_d} \\ & + \sum_{i j_c k l_c} s_{ik} tc_{j_c l_c} \log \varphi^c_{k l_c}(x_{i j_c}; \alpha_{k l_c}) + \sum_{i j_d k l_d} s_{ik} td_{j_d l_d} \log \varphi^d_{k l_d}(x_{i j_d}; \alpha_{k l_d}), \\ & - \sum_{ik} s_{ik} \log s_{ik} - \sum_{j_c l_c} tc_{j_c l_c} \log tc_{j_c l_c} - \sum_{j_d l_d} td_{j_d l_d} \log td_{j_d l_d}. \end{aligned}$$

which provides an approximation for the likelihood. The maximization of F_c is simpler to conduct and yields a maximization of the expected full-information log-likelihood. Therefore, the goal will onward be to maximize the criterion F_c.

3.1.1 The Variational Expectation Maximization algorithm

Maximizing the lower bound F_c in the mixed-data latent block model (MLBM) is performed, until convergence, in three steps:

- with regard to s, with fixed θ, tc and td, which amounts to computing

$$\hat{s}_{ik} \propto \pi_k \exp\Big(\sum_{j_c l_c} tc_{j_c l_c} \log \varphi^c_{kl_c}(x_{ij_c}, \alpha_{kl_c})\Big) \exp\Big(\sum_{j_d l_d} td_{j_d l_d} \log \varphi^d_{kl_d}(x_{ij_d}, \alpha_{kl_d})\Big) \quad (2)$$

- with regard to tc and td with fixed s and θ, which amounts to computing

$$\begin{aligned} \hat{tc}_{j_c l_c} &\propto \rho_{l_c} \exp\Big(\sum_{ik} s_{ik} \log \varphi^c_{kl_c}(x_{ij_c}, \alpha_{kl_c})\Big) \\ \text{and } \hat{td}_{j_d l_d} &\propto \rho_{l_d} \exp\Big(\sum_{ik} s_{ik} \log \varphi^d_{kl_d}(x_{ij_d}, \alpha_{kl_d})\Big), \end{aligned} \quad (3)$$

  with: $\sum_k s_{ik} = \sum_{l_c} tc_{jl_c} = \sum_{l_d} td_{jl_d} = 1$.
- with regard to θ, which amounts to computing the cluster proportions and parameters

$$\begin{aligned} \hat{\pi}_k &= \frac{\sum_i \hat{s}_{ik}}{n}\ ;\ \hat{\rho}_{l_c} = \frac{\sum_{j_c} \hat{tc}_{jl_c}}{d_c};\ \hat{\rho}_{l_d} = \frac{\sum_{j_d} \hat{td}_{jl_d}}{d_d};\ \hat{\mu}_{kl_c} = \frac{\sum_{ij_c} \hat{s}_{ik} \hat{tc}_{j_c l_c} x_{ij_c}}{\sum_i \hat{s}_{ik} \sum_{j_c} \hat{tc}_{j_c l_c}}; \\ \hat{\sigma}^2_{kl_c} &= \frac{\sum_{ij_c} \hat{s}_{ik} \hat{tc}_{j_c l_c} (x_{ij_c} - \hat{\mu}_{kl_c})^2}{\sum_i \hat{s}_{ik} \sum_{j_c} \hat{tc}_{j_c l_c}} \text{ et } \hat{\alpha}_{kl_d} = \frac{\sum_{ij_d} \hat{s}_{ik} \hat{td}_{j_d l_d} x_{ij_d}}{\sum_i \hat{s}_{ik} \sum_{j_d} \hat{td}_{j_d l_d}} \end{aligned} \quad (4)$$

In our implementation (Algorithm 1), we used $\epsilon = 10^{-5}$ as convergence constant for the inner loops, $\epsilon = 10^{-10}$ for the outer loop, and we normalized $\hat{s}$, $\hat{tc}$ and $\hat{td}$, after calculation, by taking the relative values : $\hat{s}_{ik} \leftarrow \frac{\hat{s}_{ik}}{\sum_h \hat{s}_{ih}}$ and similarly for $\hat{tc}$ and $\hat{td}$.

4 Experiments

In this section, we evaluate the proposed approach on simulated data with controlled setups. This evaluation step is necessary to measure how well the approach can uncover the true distributions from data with known parameters. To do this, we start by presenting the setups used to produce artificial data followed by an analysis of the experimental results of the proposed LBM extension. The first experiment is set to validate our implementation on uni-type data and confirm the contribution of the approach. The second experiment is set to investigate the influence of various parameters such as the number of co-clusters, the size of the data matrix and the level of overlap in the data.

Algorithm 1: : The Mixed-data Latent Block Model VEM algorithm

Require: Data $\mathbf{x}$, the number of clusters g, m_c, m_d, maximum number of iterations $maxITER$ and $InnerMaxIter$
iteration c←0
Initialization : choose $s = c^c$, $td = tc^c, td = td^c$ randomly and compute $\theta = \theta^c$ (equation (4))
while $c \leq maxITER$ **and** Unstable($Criterion$) **do**
t←0, $s^t \leftarrow s^c, tc \leftarrow tc^c, td \leftarrow td^c, \theta^t \leftarrow \theta^c$
while $t \leq InnerMaxIter$ **and** Unstable($Criterion$) **do**
For every $i = 1 : n$ and $k = 1 : g$, compute s_{ik}^{t+1} : equation (2)
For every $k = 1 : g$, $l_c = 1 : m_c$ and $l_d = 1 : m_d$, compute $\pi_k^{t+1}, \mu_{kl_c}^{t+1}, \sigma_{kl_c}^{t+1}$ et $\alpha_{kl_d}^{t+1}$: equation (4)
$Criterion \leftarrow F_c(s^{t+1}, tc, td, \theta^{t+1})$
$t \leftarrow t + 1$
end while
$s \leftarrow s^{c+1} \leftarrow s^{t-1}$, $\theta \leftarrow \theta^{c+1} \leftarrow \theta^{t-1}$
$t \leftarrow 0$
while $t \leq InnerMaxIter$ **and** Unstable($Criterion$) **do**
For every $j_c = 1 : d_c$, $j_d = 1 : d_d$, $l_c = 1 : m_c$ and $l_d = 1 : m_d$, compute $tc_{j_c l_c}^{t+1}$ and $td_{j_d l_d}^{t+1}$: equation (3)
For every $k = 1 : g$, $l_c = 1 : m_c$ and $l_d = 1 : m_d$, compute ρ_c^{t+1}, ρ_d^{t+1}, $\mu_{kl_c}^{t+1}$, $\sigma_{kl_c}^{t+1}$ and $\alpha_{kl_d}^{t+1}$: equation (4)
$Criterion \leftarrow F_c(s, tc^{t+1}, td^{t+1}, \theta^{t+1})$
$t \leftarrow t + 1$
end while
$tc \leftarrow tc^{c+1} \leftarrow tc^{t-1}$, $td \leftarrow td^{c+1} \leftarrow td^{t-1}$, $\theta \leftarrow \theta^{c+1} \leftarrow \theta^{t-1}$
$Criterion \leftarrow F_c(s, tc, td, \theta)$
$c \leftarrow c + 1$
end while
Ensure: (s, tc, td, θ)

4.1 First experiment

The purpose of this experiment is two-fold: validate our implementation and evaluate the interest of considering continuous and binary data jointly.

4.1.1 The data set

Our first data sets consist of simulated data matrices containing $g = 4$ clusters of rows, $m_c = 2$ clusters of continuous columns and $m_d = 2$ clusters of binary columns.

The particularity of this experiment lies in the fact that independently co-clustering the continuous and the binary parts of the data would only distinguish two clusters of rows but jointly, the co-clustering of the data sets should extract four clusters of rows. In this experiment, we study the effect of:

- **The size of the data matrix**: the data size is defined by the number of rows which is equal to the number of continuous columns and to the number of binary columns. We consider the sizes 25, 50, 100, 200 and 400 rows (and columns of each type) for which the resulting matrices will have 25×50, 50×100, 100×200, 200×400 and 400×800 entries respectively.
- **The level of confusion** where we study the effect of the overlap between the distributions. Here, we consider three levels of overlap (called confusion) between the co-clusters:
  - *Low*: every continuous co-cluster follows a Gaussian distribution of mean $\mu \in \{\mu_1 = 1, \mu_2 = 2\}$ and a standard deviation $\sigma = 0.25$ while a binary co-cluster follows a Bernoulli distribution of parameter $\alpha \in \{\alpha_1 = 0.2, \alpha_2 = 0.8\}$). This setup provides easily separable co-clusters since the regions of overlap between the observed values is small.
  - *Medium*: every continuous co-cluster follows a Gaussian distribution of mean $\mu \in \{\mu_1 = 1, \mu_2 = 2\}$ and a standard deviation $\sigma = 0.5$ while a binary co-cluster follows a Bernoulli distribution of parameter $\alpha \in \{\alpha_1 = 0.3, \alpha_2 = 0.7\}$. This setup provides a relatively large overlap region which should make the cluster separability harder than in the case of low confusion.
  - *High*: every continuous co-cluster follows a Gaussian distribution of mean $\mu \in \{\mu_1 = 1, \mu_2 = 2\}$ and a standard deviation $\sigma = 1$ while a binary co-cluster follows a Bernoulli distribution of parameter $\alpha \in \{\alpha_1 = 0.4, \alpha_2 = 0.6\}$. This provides a large overlap region which should make the cluster separability even more difficult.

The exact configuration of the parameters is shown in Table 1. One should note that a Gaussian mixture based co-clustering on the columns Jc_1 and Jc_2 from Table 1, would distinguish two clusters of rows by coupling $\{I_1$ and $I_3\}$, on one hand, then $\{I_2$ and $I_4\}$, on the other hand, as single row clusters. Similarly, a Bernoulli based co-clustering on the columns Jd_1 and Jd_2 should distinguish two clusters of rows by associating $\{I_1$ with $I_2\}$ and $\{I_3$ with $I_4\}$. By performing a co-clustering on the mixed data, we expect to distinguish four clusters of rows.

μ and α	Jc_1	Jc_2	Jd_1	Jd_2
I_1	μ_2	μ_1	α_2	α_1
I_2	μ_2	μ_2	α_2	α_1
I_3	μ_2	μ_1	α_2	α_2
I_4	μ_2	μ_2	α_2	α_2

Table 1: *The specification of the true parameters.*

Our experiments are performed in two steps: apply the co-clustering algorithm to the continuous data and to the binary data separately, then apply the algorithm on the mixed data.

4.1.2 Evaluation of the results

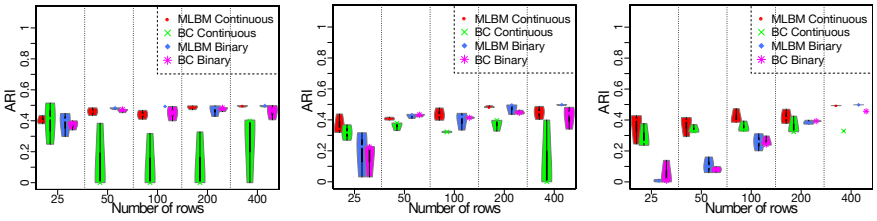
Knowing the true clusters of each row and column of the data, we choose to measure the performance of a co-clustering using the Adjusted Rand's Index ([Hubert and Arabie, 1985]) for the rows and columns. The Adjusted Rand index (ARI) is a commonly used measure of similarity between two data clusterings that can be used to measure the distance (as a probability of agreements) between the true row and column partitions and the partitions found by the co-clustering. The ARI has a maximum value of 1 for identical partitions and a minimal value of zero for independent partitions. We will thus recover and compare the ARI of rows and columns in the three cases: when co-clustering the continuous data alone, when co-clustering the binary data alone, and when co-clustering the mixed data.

For each configuration, we generate 3 data samples according to the previously illustrated parameters and we present the results in the form of violin plots. A violin plot ([Hintze and Nelson, 1998]) is a numeric data visualization method that combines the advantages of a box plots with an estimation of the probability density over the different values, which gives a better visualization of the variability of the results as well as important statistics such as the mean, the median and the extent of the measured values.

4.1.3 Validating our implementation

To validate our implementation, we applied our implementation of the co-clustering algorithm to the continuous part alone and to the binary part alone while comparing the results with those of the blockcluster package ([Bhatia et al., 2014]). Blockcluster is an R package for co-clustering binary, contingency, continuous and categorical data that implements the standard latent block models for co-clustering uni-type data.

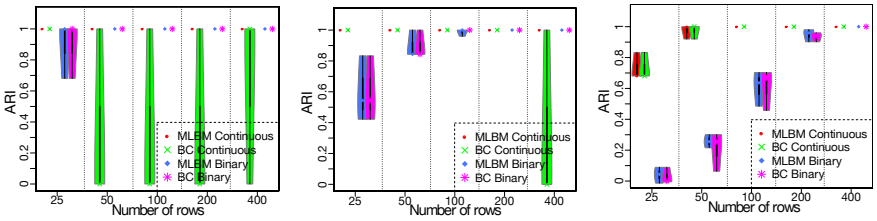
Figure 1 shows a comparison between the adjusted Rand index (of rows and columns) of the co-clustering obtained using blockcluster compared to our proposed approach. The comparison confirms that our implementation provides very comparable results, in terms of ARI and of parameter estimation, with respect to the blockcluster package in most of the cases. In particular, BC provides better ARI when co-clustering the binary data while our approach provides similar or remarkably better results when co-clustering the continuous data. However, in terms of computation time, our implementation takes at least ten times longer than the blockcluster package. This is mainly because we needed high quality in our comparison experiments,



(a) ARI of rows, Low confusion

(b) ARI of rows, Medium confusion

(c) ARI of rows, High confusion



(d) ARI of columns, Low confusion

(e) ARI of columns, Medium confusion

(f) ARI of columns, High confusion

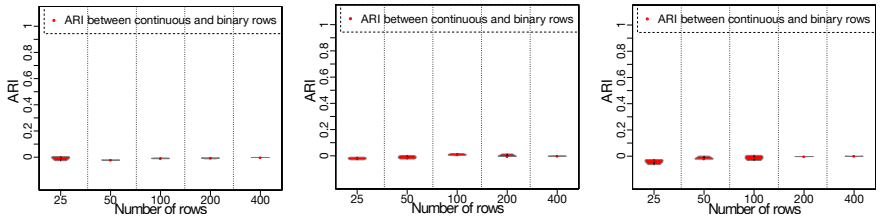
Fig. 1: *First experiment: comparing the ARI of rows and the ARI of columns (the y-axis) using our implementation (MLBM) with the blockcluster (BC) package, applied to the continuous and binary data. Compare the red plots with the green ones and the blue with the magenta. The higher the ARI values, the better.*

and therefore we focused on quality rather than computation time in our implementation (see section 5).

4.1.4 The advantage of mixed data co-clustering

One approach to co-clustering mixed data consists of performing a co-clustering on each data type then jointly analyzing the results to conclude a co-clustering like structure for the complete data. This experiment provides an example of configurations where such joint analysis remains incapable of finding the true clusters of rows.

Figure 2 compares the partition of rows found by the co-clustering of the continuous data with the partition found by the co-clustering of the binary part. Had the two co-clusterings correctly discovered the true clusters of rows, the partitions would be coherent and the ARI would approach 1, which is not the case. In fact, regardless of the data size and of the level of overlap between the distributions, the two partitions are completely independent as shown by the ARI values, which are at maximum zero. This shows that although the same row clusters are present in both data types, the joint



(a) Low confusion (b) Medium confusion (c) High confusion

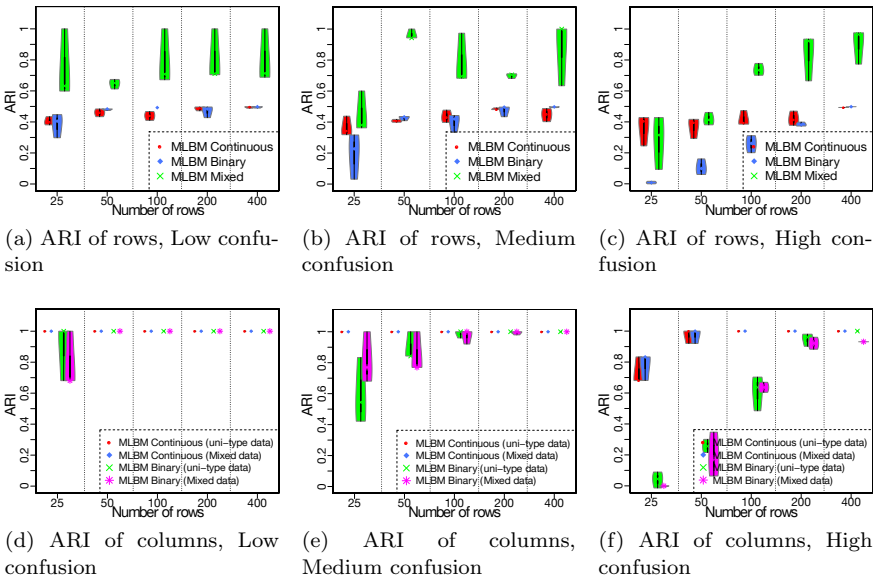
Fig. 2: *First experiment: comparing the partition of rows obtained using the continuous part alone with the partition obtained using the binary part of the data. The y axis shows the measured ARI values.*

analysis of the two independent co-clusterings does not extract the common and global structure and does not provide any additional information on the true distribution compared to a uni-type data analysis.

Given such mixed data, the correspondence between the continuous and binary partitions is virtually null. This leaves the choice open for interpreting either the continuous co-clusters or the binary ones. Our approach proposes to use the full data and provides co-clusters for which the accuracy of the row clusters is at least as good as the best of the two choices. Furthermore, mixed data co-clustering significantly improves the accuracy of the retrieved partition in the majority of the studied cases (Figure 3).

From figure 3, it is clear that, regardless of the level of overlap between the distributions and regardless of the size of the matrix, co-clustering the mixed data, instead of separately co-clustering the continuous and the binary parts, improves significantly the quality of the obtained row partition (see figures 3a, 3b, and 3c). In fact, in the worst case scenarios, mixed data co-clustering provides ARI of rows that are at least as good as the best ARI results when performing uni-type data analysis. On the other hand, the adjusted Rand indexes of columns do not necessarily improve significantly (in some cases, it does), which is expected because the configuration is set so that the clusters of columns are separable using uni-type data and the mixed analysis would not improve the performance of the clustering of columns (independence between the two data types in terms of column clusters). With respect to the data size and the level of overlap between the distributions, we notice the following:

- Influence of the data size: as the data size increases, the quantity of the data units used by the optimization algorithm increases, which facilitates the convergence of the algorithms to the true underlying distributions. This effect can be observed from the ARI values, shown in Figure 3, and mainly in the case of binary and mixed data.
- Influence of the level of confusion: as expected, when the level of confusion between the distributions increases, it becomes harder to recover the exact



(a) ARI of rows, Low confusion

(b) ARI of rows, Medium confusion

(c) ARI of rows, High confusion

(d) ARI of columns, Low confusion

(e) ARI of columns, Medium confusion

(f) ARI of columns, High confusion

Fig. 3: *First experiment: ARI of rows and ARI of columns (in the y-axis) in the continuous, binary and mixed data.*

true partition of rows. This effect is particularly visible in Figures 3f and 3c, where the high level of confusion makes the separation of the clusters difficult in the case of binary (and consequently mixed) data and particularly in small matrices.

To summarize, the joint co-clustering of the continuous and binary variables of the simulated data sets enables us to use the full data and obtain considerably better accuracy, compared to an independent analysis by data type. The results of the co-clustering (both uni-type and mixed) are at their best when the level of confusion is low or the data matrix is big. With respect to the level of confusion, this behavior is expected since the true structure of the data is well separable. In fact, the level of confusion simulates the overlap between the distributions. Therefore, the higher the overlap, the data will contain more observations with relatively equal probabilities to belong to either of the distributions. Hence, a decrease in the accuracy of the clustering as it is measured over all the observations. The effect of bigger matrices can be explained by the fact that the more data is present, the more iterations are required by the algorithm which improves the quality of the estimated parameters.

However, this is a well known phenomenon in the standard the standard LBM context. For example, in [Govaert and Nadif, 2013], the authors note that given the same number of co-clusters in the data, the classification error

rate depends not only on the parameters but also on the size of the data matrix (the Bayes classification risk decreases with the size of the data). Also, [Mariadassou and Matias, 2015] show that, when the estimated parameters converge to the true parameters, the recovered partitions will converge to the true partitions when the size of the data becomes sufficiently large. Using the mixed data latent block model, the accuracy of the estimated parameters is remarkable which reinforces the hypothesis that, as in the standard latent block models, given enough data, our approach would converge to the true partitions. Table 2 shows examples of the estimated parameters using the mixed data latent block model MLBM on the data containing 100 rows.

	(True μ, estimated μ)		(True σ, estimated σ)		(True α, estimated α)	
Low confusion	Jc_1	Jc_2	Jc_1	Jc_2	Jd_2	Jd_1
	I_1 (2, 2.001)	(1, 1.004)	(0.25, 0.246)	(0.25, 0.255)	(0.2, 0.208)	(0.8, 0.790)
	I_4 (2, 1.992)	(2, 1.013)	(0.25, 0.250)	(0.25, 0.239)	(0.8, 0.804)	(0.8, 0.822)
	I_2 (2, 2.000)	(2, 2.005)	(0.25, 0.251)	(0.25, 0.248)	(0.2, 0.496)	(0.8, 0.811)
	I_3 (2, 1.988)	(1, 0.975)	(0.25, 0.266)	(0.25, 0.259)	(0.8, 0.797)	(0.8, 0.758)
	(True μ, estimated μ)		(True σ, estimated σ)		(True α, estimated α)	
Medium confusion	Jc_2	Jc_1	Jc_2	Jc_1	Jd_1	Jd_2
	I_2 (2, 1.981)	(2, 1.985)	(0.50, 0.520)	(0.50, 0.497)	(0.7, 0.718)	(0.3, 0.297)
	I_1 (1, 1.023)	(2, 1.995)	(0.50, 0.502)	(0.50, 0.493)	(0.7, 0.695)	(0.3, 0.315)
	I_4 (2, 1.990)	(2, 2.010)	(0.50, 0.505)	(0.50, 0.505)	(0.7, 0.704)	(0.7, 0.685)
	I_3 (1, 1.013)	(2, 1.962)	(0.50, 0.501)	(0.50, 0.504)	(0.7, 0.700)	(0.7, 0.674)
	(True μ, estimated μ)		(True σ, estimated σ)		(True α, estimated α)	
High confusion	Jc_2	Jc_1	Jc_2	Jc_1	Jd_2	Jd_1
	I_2 (2, 1.994)	(2, 2.042)	(1.00, 1.016)	(1.00, 0.989)	(0.4, 0.351)	(0.6, 0.597)
	I_1 (1, 0.998)	(2, 1.987)	(1.00, 1.018)	(1.00, 0.979)	(0.4, 0.399)	(0.6, 0.594)
	I_4 (2, 2.005)	(2, 2.008)	(1.00, 0.999)	(1.00, 1.001)	(0.6, 0.591)	(0.6, 0.590)
	I_3 (1, 1.012)	(2, 2.017)	(1.00, 0.992)	(1.00, 1.008)	(0.6, 0.631)	(0.6, 0.616)

Table 2: *Examples of the estimated parameters for the* 100 *rows data.*

4.2 Second experiment

The objective of this experiment is to study the impact of the number of co-clusters, the size of the data and the level of confusion between the distributions.

4.2.1 The data set

To study the influence of the number of co-clusters, the data sets are generated using the following parameters:

- **The number of co-clusters**: we choose three different partitions $g \times (m_c + m_d)$ of the original data matrix : $2 \times (2 + 2)$, $3 \times (3 + 3)$, and $4 \times (4 + 4)$.

- **The size of the data**: the size of the data is defined by the number of rows and the total number of columns. For this experiment, we choose the sizes 25, 50, 100, 200 and 400 for rows. For the number of columns, we distinguish two different configurations:
  - square matrices: the number of columns of each type is equal to the number of rows.
  - rectangular matrices: we set the number of columns (of each type) to 5, 10, and 20.
- **The level of confusion**: similarly to the first experiment, we consider three levels of overlap between the distributions: *Low* (with Gaussian means $\mu \in \{p_1 = 1, p_2 = 2\}$, Gaussian standard deviations $\sigma = 0.25$ and Bernoulli parameters $\alpha \in \{p_1 = 0.2, p_2 = 0.8\}$), *Medium* ($\mu \in \{p_1 = 1, p_2 = 2\}$, $\sigma = 0.5$ and $\alpha \in \{p_1 = 0.3, p_2 = 0.7\}$) and *High* ($\mu \in \{p_1 = 1, p_2 = 2\}$, $\sigma = 1$ and $\alpha \in \{p_1 = 0.4, p_2 = 0.6\}$).

μ or α	J_1	J_2
I_1	p_1	p_1
I_2	p_1	p_2

μ or α	J_1	J_2	J_3
I_1	p_1	p_2	p_1
I_2	p_1	p_2	p_2
I_3	p_1	p_1	p_1

μ or α	J_1	J_2	J_3	J_4
I_1	p_2	p_1	p_2	p_1
I_2	p_2	p_1	p_2	p_2
I_3	p_2	p_2	p_2	p_2
I_4	p_2	p_1	p_1	p_1

Table 3: *The true parameter specification with* $2 \times (2 + 2)$, $3 \times (3 + 3)$ *and* $4 \times (4 + 4)$ *co-clusters.*

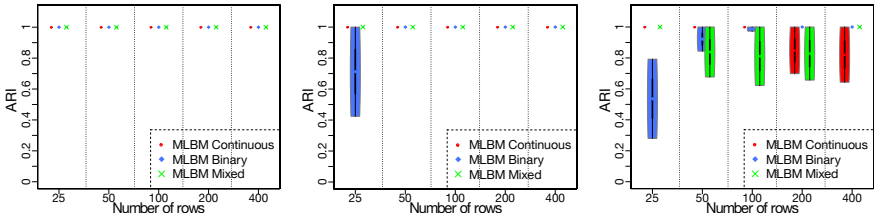
The specification of the co-clusters and their configuration are shown in Table 3. Similarly to the first experiment, we generate 3 samples of each data configuration according to its parameters and we present the resulting ARI in the form of violin plots. To present the co-clustering results, we distinguish between square matrices and rectangular ones.

4.2.2 The co-clustering results: square matrices

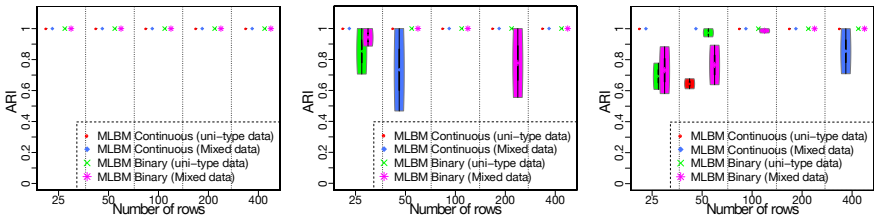
Although each of the continuous and binary parts of the data can be sufficient to extract the underlying structure of the data, we notice that, as in the first experiment, jointly co-clustering the continuous and binary data clearly improves the performance of the co-clustering.

Figures 4, 5 and 6 show the adjusted Rand indexes of rows and columns by level of confusion and with respect to the various parameters, in the case of continuous, binary and mixed data co-clustering.

From the ARI plots (Figure 4 and Figure 5 in particular), the first noticeable result is that the binary part of the data is sensitive to the size of the data, to the number of co-clusters and to the level of confusion while the

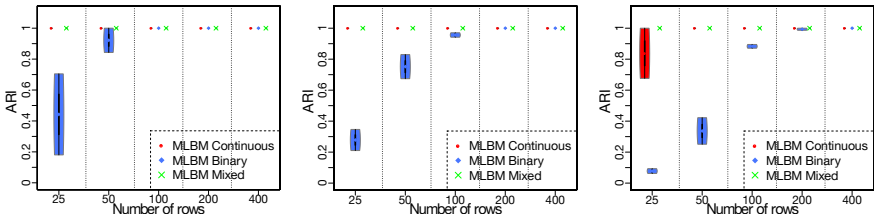


(a) ARI of rows, $2 \times (2+2)$ (b) ARI of rows, $3 \times (3+3)$ (c) ARI of rows, $4 \times (4+4)$

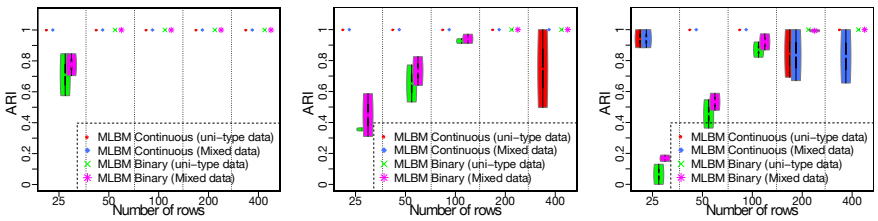


(d) ARI of columns, $2 \times (2+2)$ (e) ARI of columns, $3 \times (3+3)$ (f) ARI of columns, $4 \times (4+4)$

Fig. 4: *Second experiment (Low confusion): ARI of rows and ARI of columns (in the y-axis) in the case of continuous, binary and mixed data.*

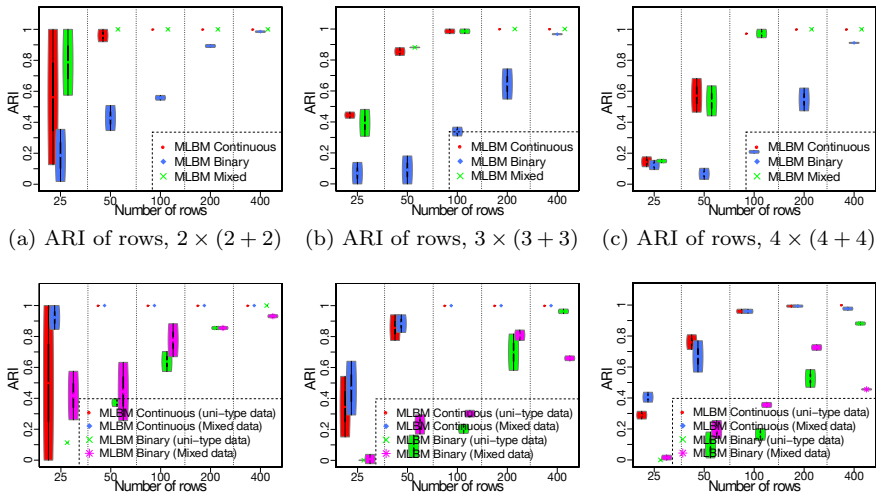


(a) ARI of rows, $2 \times (2+2)$ (b) ARI of rows, $3 \times (3+3)$ (c) ARI of rows, $4 \times (4+4)$



(d) ARI of columns, $2 \times (2+2)$ (e) ARI of columns, $3 \times (3+3)$ (f) ARI of columns, $4 \times (4+4)$

Fig. 5: *Second experiment (Medium confusion): ARI of rows and ARI of columns (the y-axis) in the case of continuous, binary and mixed data.*



(a) ARI of rows, $2 \times (2+2)$ (b) ARI of rows, $3 \times (3+3)$ (c) ARI of rows, $4 \times (4+4)$

(d) ARI of columns, $2 \times (2+2)$ (e) ARI of columns, $3 \times (3+3)$ (f) ARI of columns, $4 \times (4+4)$

Fig. 6: *Second experiment (High confusion): ARI of rows and ARI of columns (the y-axis) in the case of continuous, binary and mixed data.*

continuous part is generally more stable and is mostly influenced only by the number of co-clusters and the size of the data.

- Influence of the number of co-clusters: given the same data size and the same level of overlap between the clusters, we notice (see Figure 4) that as the number of co-clusters increases, the extraction of the true partition (both in terms of rows and column clusters) becomes harder. This effect is observed in particular in the case of binary variables as the variability of the results is greater when the number of co-clusters becomes high. However, this variability is less drastic in the case of continuous and mixed data (see Figures 5a , 5b, and 5c for example). The greater the number of clusters, the more data is required for the true partition to be found.
- Influence of the data size: the global performances of the co-clustering of uni-type data (which we have established is equivalent to the standard LBM co-clustering) confirms (as established in section 4.1.4) that the co-clustering performs better as the data size increases. Additionally, we notice that the continuous part of the data is always easier to co-cluster than the binary part. This is almost regardless of the data size (except in the case of large number of co-clusters: $4 \times (4+4)$). The binary part on the other hand performs particularly worse for small matrices. In summary, the best partitioning of the mixed co-clusters is obtained, regardless of the number of co-clusters and the level of confusion, with medium to large matrices.

- Influence of the level of confusion: the co-clustering of the mixed data performs as expected with respect to the level of confusion. The higher the confusion, the more difficult is the extraction of the true partition of the rows, particularly in the case of small matrices (compare for example the Figures 4a, 5a, and 6a). On the contrary, even when the level of confusion is high, the quality of the recovered co-clusters improves with the size of the data (see the evolution of the ARI values in Figure 6).

4.2.3 The co-clustering results: rectangular matrices

Figure 7 shows the adjusted Rand indexes of rows by level of confusion and with respect to the various parameters, in the case of rectangular matrices and $2 \times (2 + 2)$ co-clusters.

From this experiment, we notice that even for rectangular matrices, the same conclusions are valid. In particular, the proposed approach extracts the true structure of the data in the case of low confusion. As the level of overlap between the co-clusters increases, the co-clustering of the binary part becomes less accurate both in the case of a standard LBM on uni-type data and the case of mixed data. Finally, the bigger the data size, the more accurate is the co-clustering both using uni-type and mixed data. As with the square matrices, an improvement in the ARI of columns is also noticed when using mixed data. The same conclusions are valid for the configurations containing $3 \times (3 + 3)$ and $4 \times (4 + 4)$ co-clusters.

5 Discussion

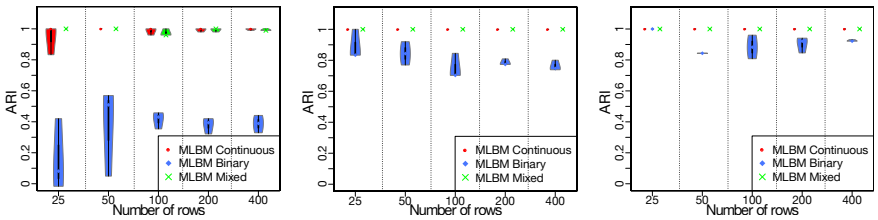
When applying the co-clustering algorithm on uni-type data, we noticed some optimization problems. Firstly, the algorithm converges to a local optimum which corresponds, very often, to a unique cluster of rows and a unique cluster of columns. We have thus addressed the problem by forcing a minimal number of iterations (the c parameter in Algorithm 1) which considerably enhanced the quality of the optimization results.

μ or α	J_1	J_2
I_1	p_1	p_2
I_2	p_2	p_1

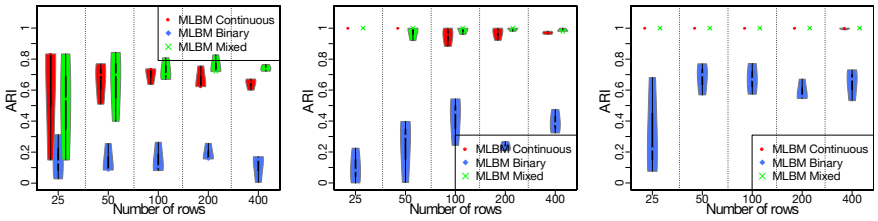
μ or α	J_1	J_2	J_3
I_1	p_1	p_1	p_2
I_2	p_1	p_2	p_1
I_3	p_2	p_1	p_1

μ or α	J_1	J_2	J_3	J_4
I_1	p_1	p_1	p_1	p_2
I_2	p_1	p_1	p_2	p_1
I_3	p_1	p_2	p_1	p_1
I_4	p_2	p_1	p_1	p_1

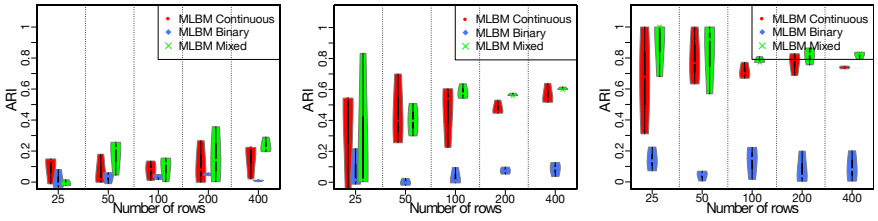
Table 4: *The true specification of the co-clusters in a symmetric configuration with* $2 \times (2 + 2)$*,* $3 \times (3 + 3)$ *and* $4 \times (4 + 4)$ *co-clusters.*



(a) Low confusion, 5 columns (b) Low confusion, 10 columns (c) Low confusion, 20 columns



(d) Medium confusion, 5 columns (e) Medium confusion, 10 columns (f) Medium confusion, 20 columns



(g) High confusion, 5 columns (h) High confusion, 10 columns (i) High confusion, 20 columns

Fig. 7: *Second experiment ($2 \times (2+2)$ co-clusters): ARI of rows (the y-axis) in the case of continuous, binary and mixed data.*

However, the algorithms (both our approach and the blockcluster package) do not perform the same way when the marginal parameters are equal per cluster or when they are different. To study this effect, we have considered a second configuration (call it the *symmetric* case) where the marginal parameters are equal. Table 4 shows an example of the parameter specification of such configurations. In the *symmetric* configuration, where the marginal parameters are equal, the problem of cluster separability becomes intrinsically difficult (especially for square matrices) and the optimization algorithm tends to have trouble getting out of the zone of the local optimum corresponding to one single cluster of rows and one single cluster of columns, in which it falls since the very first iterations. To solve this problem, we required the algorithm to start with small steps when computing the assignments to the

clusters (s, tc and td) without letting the criterion fully stabilize, then after few first steps in this initial phase, we iterate until criterion stabilization. This strategy provides better solutions in the case of binary data but results in no notable improvement in some continuous cases. As mentioned earlier, because of this focus on obtaining high quality results, our implementation takes at least ten times longer than the blockcluster package but provides more accurate row and column partitions and more accurate parameter estimation. Table 5 shows a comparative example of the means computation time for the rectangular matrix containing 100 rows and $2 \times (2 + 2)$ co-clusters.

Number of columns	→	5 columns		10 columns		20 columns	
Level of overlap	measure\method	MLBM	BC	MLBM	BC	MLBM	BC
Low confusion	mean	2.97	0.01	4.3	0.016	8.31	0.03
	sd	0.11	0.005	0.4	0.005	0.6	0.005
Medium confusion	mean	8.01	0.01	5.5	0.01	8.4	0.01
	sd	1.2	0.01	0.2	0	0.3	0
High confusion	mean	15	0.04	16.4	0.01	25.9	0.01
	sd	12	0.01	6.4	0.01	5.2	0.01

Table 5: *Example of the computation time (in seconds).*

6 Conclusion and future work

In this article, we have proposed an extension of the latent block models to the co-clustering of mixed type data. The experiments show the capability of the approach to estimate the true model parameters, extract the true distributions from simulated data, and provide better quality results when the complete data set is used rather than separately co-clustering the continuous or binary parts. The proposed approach comes as a natural extension of the LBM based co-clustering and performs a co-clustering of mixed data in the same way that a standard LBM based co-clustering applies to uni-type data.

On the course of our experiments, we have noticed that for the data sets with equal marginal parameters, both our algorithm and the state of the art algorithm implemented in the package blockcluster tend to fall in a local optimum. This is a limitation to the latent block based methods for co-clustering, mainly in the context of an exploratory analysis where the true underlying distributions are unknown.

In our future works, we aim to extend the approach to the case of categorical data and beyond binary data and to study the option of BIC based regularization to automatically infer the number of clusters of rows and the number of clusters of columns.

References

Bhatia et al., 2014. Bhatia, P., Iovleff, S., and Govaert, G. (2014). blockcluster: An r package for model based co-clustering. working paper or preprint : `https://hal.inria.fr/hal-01093554`.

Bock, 1979. Bock, H. (1979). Simultaneous clustering of objects and variables. In *E. Diday (ed) Analyse des données et Informatique*, page 187–203. INRIA.

Brault and Lomet, 2015. Brault, V. and Lomet, A. (2015). Revue des méthodes pour la classification jointe des lignes et des colonnes d'un tableau. *Journal de la Société Française de Statistique*, 156(3):27–51.

Cheng and Church, 2000. Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, volume 8, pages 93–103. AAAI Press.

Dhillon et al., 2003. Dhillon, I. S., Mallela, S., and Modha, D. S. (2003). Information-theoretic co-clustering. In *Proceedings of the ninth international conference on Knowledge discovery and data mining*, pages 89–98. ACM Press.

Good, 1965. Good, I. J. (1965). Categorization of classification. In *Mathematics and Computer Science in Biology and Medicine*, pages 115–125. Her Majesty's Stationery Office, London.

Govaert and Nadif, 2003. Govaert, G. and Nadif, M. (2003). Clustering with block mixture models. *Pattern Recognition*, 36(2):463–473.

Govaert and Nadif, 2008. Govaert, G. and Nadif, M. (2008). Block clustering with Bernoulli mixture models : Comparison of different approaches. *Computational Statistics and Data Analysis*, 52(6):3233–3245.

Govaert and Nadif, 2013. Govaert, G. and Nadif, M. (2013). *Co-Clustering*. ISTE Ltd and John Wiley & Sons Inc.

Hartigan, 1975. Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.

Hintze and Nelson, 1998. Hintze, J. L. and Nelson, R. D. (1998). Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184.

Hubert and Arabie, 1985. Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

Mariadassou and Matias, 2015. Mariadassou, M. and Matias, C. (2015). Convergence of the groups posterior distribution in latent or stochastic block models. *Bernoulli*, 21(1):537–573.

McParland and Gormley, 2016. McParland, D. and Gormley, I. C. (2016). Model based clustering for mixed data: clustmd. *Advances in Data Analysis and Classification. Springer*, 10(2):155–169.

Bibliography

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723. (Cited in pages 27 and 38.)

Amant, R. S. and Cohen, P. R. (1995). Issues in automating exploratory data analysis. (Cited in pages 1 and 10.)

Anagnostopoulos, A., Dasgupta, A., and Kumar, R. (2008). Approximation algorithms for co-clustering. In *Proceedings of the Twenty-seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '08, pages 201–210, New York, NY, USA. ACM. (Cited in page 21.)

Anderberg, M. R. (1973). *Cluster Analysis for Applications*. ELSEVIER. (Cited in pages 9 and 11.)

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., and Modha, D. S. (2007). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. *J. Mach. Learn. Res.*, 8:1919–1986. (Cited in page 26.)

Beh, E. and Lombardo, R. (2014). *Correspondence Analysis: Theory, Practice and New Strategies*. Wiley. (Cited in page 10.)

Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(7):719–725. (Cited in pages 27 and 38.)

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg. (Cited in pages 8 and 12.)

Bouchareb, A., Boullé, M., Clérot, F., and Rossi, F. (2017a). Application du coclustering à l'analyse exploratoire d'une table de données. In *17ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2017*, volume RNTI-E-33, pages 177–188. (Cited in page 3.)

Bouchareb, A., Boullé, M., Clérot, F., and Rossi, F. (2018a). Co-clustering based exploratory analysis of mixed-type data tables. In Pinaud, B., Gandon, F., Bisson, G., and Guillet, F., editors, *Accepted for publication in Advances in Knowledge Discovery and Management Vol. 8 (AKDM-8)*. Springer.

Bouchareb, A., Boullé, M., Clérot, F., and Rossi, F. (2018b). Model based co-clustering of mixed numerical and binary data. In Pinaud, B., Gandon, F., Bisson, G., and Guillet, F., editors, *Accepted for publication in Advances in Knowledge Discovery and Management Vol. 8 (AKDM-8)*. Springer.

Bouchareb, A., Boullé, M., and Rossi, F. (2017b). Co-clustering de données mixtes à base des modèles de mélange. In *17ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2017, 24-27 Janvier 2017, Grenoble, France*, pages 141–152. (Cited in page 4.)

Bouchareb, A., Boullé, M., Rossi, F., and Clérot, F. (2018c). Un modèle bayésien de co-clustering de données mixtes. In *Extraction et Gestion des Connaissances, EGC 2018, Paris, France, January 23-26, 2018*, pages 275–280.

Boullé, M. (2007). *Recherche d'une représentation des données efficace pour la fouille de grandes bases de données.* Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications. (Cited in pages 34 and 35.)

Boullé, M. (2011). Data grid models for preparation and modeling in supervised learning. In Guyon, I., Cawley, G., Dror, G., and Saffari, A., editors, *Hands-On Pattern Recognition: Challenges in Machine Learning*, pages 99–130. Microtome Publishing. (Cited in pages 37, 42, 64, and 87.)

Brault, V. (2014). *Estimation et sélection de modèle pour le modèle des blocs latents.* Thèse de doctorat, Université Paris-Sud. (Cited in page 27.)

Brault, V. and Lomet, A. (2015). Revue des méthodes pour la classification jointe des lignes et des colonnes d'un tableau. *Journal de la Société Française de Statistique*, 156(3):27–51. (Cited in page 18.)

Brault, V. and Mariadassou, M. (2015). Co-clustering through latent bloc model : a review. *Journal de la Société Française de Statistique*, 156(3). Special Issue on Networks and Statistics. (Cited in page 18.)

Buono, N. D. and Pio, G. (2015). Non-negative matrix tri-factorization for co-clustering: An analysis of the block matrix. *Information Sciences*, 301:13–26. (Cited in page 23.)

Cano, C., Adarve, L., López, J., and Blanco, A. (2007). Possibilistic approach for biclustering microarray data. *Computers in Biology and Medicine*, 37(10):1426–1436. (Cited in page 25.)

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In *EWSL*. (Cited in page 10.)

Charrad, M. and Ahmed, M. B. (2011). Simultaneous clustering: A survey. *4th International Conference of PReMI*. (Cited in page 18.)

Chen, Y.-C., Wheeler, T. A., and Kochenderfer, M. J. (2017). Learning discrete bayesian networks from continuous data. *J. Artif. Int. Res.*, 59(1):103–132. (Cited in page 10.)

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, volume 8, pages 93–103. AAAI Press. (Cited in pages 19, 20, and 21.)

Cho, H. and Dhillon, I. S. (2008). Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(3):385–400. (Cited in page 21.)

Cho, H., Dhillon, I. S., Guan, Y., and Sra, S. (2004). Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 114–125. (Cited in pages 20 and 21.)

Cunningham, J. P. and Ghahramani, Z. (2015). Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16:2859–2900. (Cited in page 12.)

D'Enza, A. I. and Greenacre, M. (2012). Multiple correspondence analysis for the quantification and visualization of large categorical data sets. In Di Ciaccio, A., Coli, M., and Angulo Ibanez, J. M., editors, *Advanced Statistical Methods for the Analysis of Large Data-Sets*, pages 453–463, Berlin, Heidelberg. Springer Berlin Heidelberg. (Cited in page 50.)

Dhillon, I. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *the 7th ACM SIGKDD*, KDD '01, pages 269–274, New York, NY, USA. ACM. `http://doi.acm.org/10.1145/502512.502550`. (Cited in pages 24, 25, and 51.)

Dhillon, I. S., Mallela, S., and Modha, D. S. (2003). Information-theoretic co-clustering. In *Proceedings of the ninth international conference on Knowledge discovery and data mining*, pages 89–98. ACM Press. (Cited in pages 26, 33, and 34.)

Di Ciaccio, A., Coli, M., and Ibanez, J. M. A. (2012). *Advanced Statistical Methods for the Analysis of Large Data-Sets: Studies in Theoretical and Applied Statistics*. Springer Publishing Company, Incorporated. (Cited in page 50.)

Ding, C., Li, T., Peng, W., and Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 126–135, New York, NY, USA. ACM. (Cited in page 23.)

Eren, K., Deveci, M., Küçüktunç, O., and Çatalyürek, Ü. V. (2013). A comparative analysis of biclustering algorithms for gene expression data. *Briefings in bioinformatics*, 14 3:279–92. (Cited in page 42.)

Everitt, B. S., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. Wiley. (Cited in pages 1 and 10.)

Fagerland, M., Lydersen, S., and Laake, P. (2017). *Statistical Analysis of Contingency Tables*. New York: Chapman and HallCRC. (Cited in page 31.)

Fayyad, U. M. and Irani, K. B. (1993). Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *13th International Joint Conference on Uncertainly in Artificial Intelligence(IJCAI93)*, pages 1022–1029. Morgan Kaufmann. (Cited in page 11.)

Fessant, F., Benkhelif, T., and Clérot, F. (2017). Anonymiser des données multidimensionnelles à l'aide du coclustering. In *17ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2017, 24-27 Janvier 2017, Grenoble, France*, pages 153–164. (Cited in page 121.)

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188. (Cited in pages 12, 52, and 93.)

Foss, A., Markatou, M., Ray, B., and Heching, A. (2016). A semiparametric method for clustering mixed data. *Machine Learning*, 105(3):419–458. (Cited in page 11.)

Frank, E. and Witten, I. H. (1999). Making better use of global discretization. In *Proceeding of 16th International Conference on Machine Learning, Bled, Slovenia*, pages 115–123. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (Cited in page 10.)

Friedman, N. and Goldszmidt, M. (1996). Discretizing continuous attributes while learning bayesian networks. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, pages 157–165, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. (Cited in pages 10 and 11.)

Getz, G., Levine, E., and Domany, E. (2000). Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences*, 97(22):12079–12084. (Cited in page 17.)

Good, I. J. (1965). Categorization of classification. In *Mathematics and Computer Science in Biology and Medicine*, pages 115–125. Her Majesty's Stationery Office, London. (Cited in page 13.)

Govaert and Nadif, M. (2010). Latent block model for contingency table. *Communications in Statistics, Theory and Methods*, 39(3):416 – 425. (Cited in page 33.)

Govaert, G. (1983). *Classification croisée*. Thèse d'état, Université Paris 6, France. (Cited in pages 17, 21, 23, and 32.)

Govaert, G. (1995). Simultaneous clustering of rows and columns. *Control and Cybernetics*. (Cited in page 17.)

Govaert, G. and Nadif, M. (2003). Clustering with block mixture models. *Pattern Recognition*, 36(2):463–473. (Cited in pages 26 and 27.)

Govaert, G. and Nadif, M. (2007). Clustering of contingency table and mixture model. *European Journal of Operational Research*, 183(3):1055–1066. (Cited in page 27.)

Govaert, G. and Nadif, M. (2008). Block clustering with Bernoulli mixture models: Comparison of different approaches. *Computational Statistics and Data Analysis*, 52(6):3233–3245. (Cited in pages 26 and 27.)

Govaert, G. and Nadif, M. (2009). Un modèle de mélange pour la classification croisée d'un tableau de données continues. *CAP'09, 11e conférence sur l'apprentissage artificiel*. (Cited in page 27.)

Govaert, G. and Nadif, M. (2013). *Co-Clustering*. ISTE Ltd and John Wiley & Sons Inc. (Cited in page 32.)

Greenacre, M. J. and Blasius, J. (2006). *Multiple correspondence analysis and related methods*. Boca Raton: Chapman & Hall/CRC. (Cited in page 10.)

Guigourès, R., Boullé, M., and Rossi, F. (2015a). Discovering patterns in time-varying graphs: a triclustering approach. *Advances in Data Analysis and Classification*, pages 1–28. (Cited in page 46.)

Guigourès, R., Boullé, M., and Rossi, F. (2015b). Discovering patterns in time-varying graphs: a triclustering approach. *Advances in Data Analysis and Classification*, pages 1–28. (Cited in page 87.)

Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129. (Cited in pages 18, 20, and 21.)

Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA. (Cited in page 13.)

Ho, K. M. and Scott, P. D. (1997). Zeta: A global method for discretization of continuous variables. In *KDD*. (Cited in page 10.)

Ishiguro, K., Ueda, N., and Sawada, H. (2012). Subset infinite relational models. In Lawrence, N. D. and Girolami, M., editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 547–555, La Palma, Canary Islands. PMLR. (Cited in page 29.)

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323. (Cited in page 11.)

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., and Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 381–388. AAAI Press. (Cited in page 29.)

Keribin, C., Brault, V., Celeux, G., and Govaert, G. (2013). Estimation and selection for the latent block model on categorical data. Research Report RR-8264, INRIA. (Cited in page 27.)

Kluger, Y., Basri, R., Chang, J. T., and Gerstein, M. (2003). Spectral biclustering of microarray cancer data: Co-clustering genes and conditions. *Genome Research*, 13:703–716. (Cited in pages 17, 25, and 51.)

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 202–207. AAAI Press. (Cited in pages 100 and 110.)

Kuhn, H. W. and Yaw, B. (1955). The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97. (Cited in page 62.)

la Torre, F. D. (2012). A least-squares framework for component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1041–1055. (Cited in page 12.)

Laclau, C., Redko, I., Matei, B., Bennani, Y., and Brault, V. (2017). Co-clustering through Optimal Transport. In *34th International Conference on Machine Learning*, volume 70 of *Proceedings of the 34th International Conference on Machine Learning*, pages 1955–1964, Sydney, Australia. Proceedings of Machine Learning Research. (Cited in pages 31 and 34.)

Lee, D. D. and Seung, H. S. (2011). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562. (Cited in pages 12 and 22.)

Lerman, I. and Leredde, H. (1977). La méthode des pôles d'attraction. *Journées Analyse des Données et Informatique.* (Cited in page 16.)

Lichman, M. (2013). UCI machine learning repository. `http://archive.ics.uci.edu/ml`. (Cited in pages 51, 52, 57, 93, 100, and 110.)

Liu, H., Hussain, F., Tan, C. L., and Dash, M. (2002). Discretization: An enabling technique. *Data Min. Knowl. Discov.*, 6(4):393–423. (Cited in pages 10 and 11.)

Long, B., Zhang, Z. M., and Yu, P. S. (2005). Co-clustering by block value decomposition. *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. (Cited in page 23.)

Madeira, S. and Oliveira, A. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. (Cited in pages 16, 18, 20, and 42.)

Maimon, O. and Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook*. Springer Publishing Company, Incorporated, 2nd edition. (Cited in pages 8, 10, 11, and 12.)

Mansinghka, V., Shafto, P., Jonas, E., Petschulat, C., Gasner, M., and Tenenbaum, J. (2016). Crosscat: A fully bayesian nonparametric method for analyzing heterogeneous, high dimensional data. *Journal of Machine Learning Research*, 17(138):1–49. (Cited in pages 30, 38, 91, 100, and 120.)

Mansinghka, V. K. (2017). *MIT Probabilistic Computing Project:CrossCat*. (accessed last: October 3, 2018). (Cited in page 117.)

McLachlan, G. and Krishnan, T. (2008). *The EM Algorithm and Extensions*. Wiley series in probability and statistics. Wiley. (Cited in page 27.)

Mechelen, I. V., Bock, H.-H., and Boeck, P. D. (2004). Two-mode clustering methods: astructuredoverview. *Statistical Methods in Medical Research*, 13(5):363–394. (Cited in page 15.)

Meeds, T. and Roweis, S. (2007). Nonparametric bayesian biclustering. Technical report, Department of Computer Science, University of Toronto. (Cited in pages 29 and 30.)

Mittal, A. and Cheong, L.-F. (2002). Employing discrete bayes error rate for discretization and feature selection tasks. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 298–305. (Cited in page 10.)

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press. (Cited in pages 8 and 12.)

Nadif, M. and Govaert, G. (2005). Block clustering of contingency table and mixture model. In *Proceedings of the 6th International Conference on Advances in Intelligent Data Analysis*, IDA'05, pages 249–259, Berlin, Heidelberg. Springer-Verlag. (Cited in page 33.)

Padilha, V. A. and Campello, R. J. G. B. (2017). A systematic comparative evaluation of biclustering techniques. *BMC Bioinformatics*, 18(1):55. (Cited in page 18.)

Pitman, J. (2002). Combinatorial stochastic processes. Technical report, Dept. of Statistics. UC, Berkeley. (Cited in page 29.)

Pontes, B., Giráldez, R., and Aguilar-Ruiz, J. S. (2015). Biclustering on expression data: A review. *Journal of Biomedical Informatics*, 57:163 – 180. (Cited in pages 18 and 25.)

Rencher, A. C. (2002). *Methods of Multivariate Analysis*. Wiley Series in Probability and Statistics. Wiley. (Cited in pages 11 and 12.)

Saporta, G. (2006). *Probabilités, analyse des données et statistique*. Editions Technip. (Cited in pages 12 and 49.)

Scholkopf, B. and Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA. (Cited in page 12.)

Schölkopf, B. and Smola, A. J. (2002). Learning with kernels: Support vector machines, regularization, optimization, and beyond. *IEEE Transactions on Neural Networks*, 16:781–781. (Cited in page 8.)

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464. (Cited in pages 27 and 38.)

Shan, H. and Banerjee, A. (2008). Bayesian co-clustering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 530–539, Washington, DC, USA. IEEE Computer Society. `http://dx.doi.org/10.1109/ICDM.2008.91`. (Cited in pages 17, 27, and 28.)

Shan, H. and Banerjee, A. (2010). Residual bayesian co-clustering for matrix approximation. In *SDM*. (Cited in page 26.)

Snijders, T. A. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100. (Cited in page 26.)

Sun, L., Ji, S., and Ye, J. (2009). A least squares formulation for a class of generalized eigenvalue problems in machine learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 977–984, New York, NY, USA. ACM. (Cited in page 12.)

Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 Suppl 1:S136–44. (Cited in page 18.)

Tanay, A., Sharan, R., and Shamir, R. (2005). Biclustering algorithms: A survey. In *In Handbook of Computational Molecular Biology Edited by: Aluru S. Chapman & Hall/CRC Computer and Information Science Series*. (Cited in page 18.)

Tang, C., Zhang, L., Zhang, A., and Ramanathan, M. (2001). Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In *Proceedings 2nd Annual IEEE International Symposium on Bioinformatics and Bioengineering (BIBE 2001)*, pages 41–48. (Cited in page 17.)

Tishby, N., Pereira, F., and Bialek, W. (1999). The information bottleneck method. *Invited paper to The 37th annual Allerton Conference on Communication, Control, and Computing*. (Cited in page 17.)

Wang, P., Domeniconi, C., and Laskey, K. B. (2009). Latent dirichlet bayesian co-clustering. In Buntine, W., Grobelnik, M., Mladenić, D., and Shawe-Taylor, J., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 522–537, Berlin, Heidelberg. Springer Berlin Heidelberg. (Cited in pages 26 and 28.)

Wang, P., Domeniconi, C., Rangwala, H., and Laskey, K. B. (2012). Feature enriched nonparametric bayesian co-clustering. In *PAKDD*. (Cited in page 29.)

Xu, Z., Tresp, V., Yu, K., and Kriegel, H.-P. (2006). Infinite hidden relational models. *CoRR*, abs/1206.6864. (Cited in page 29.)

Yang, J., Wang, H., Wang, W., and Yu, P. (2003). Enhanced biclustering on expression data. In *Third IEEE Symposium on Bioinformatics and Bioengineering, 2003. Proceedings.*, pages 321–327. (Cited in page 20.)

Yang, W. H., Dai, D. Q., and Yan, H. (2011). Finding correlated biclusters from gene expression data. *IEEE Transactions on Knowledge and Data Engineering*, 23(4):568–584. (Cited in page 25.)

Yoo, J. and Choi, S. (2010). Orthogonal nonnegative matrix tri-factorization for co-clustering : Multiplicative updates on stiefel manifolds. *Information processing & management*. (Cited in pages 17, 22, and 23.)