Design of Artificial Data Tables for Co-Clustering Analysis

A. Lomet, G. Govaert, Y. Grandvalet

Université de Technologie de Compiègne – CNRS
Heudiasyc, UMR 7253 France

Abstract

Co-clustering aims at simultaneously partitioning the rows and columns of a data table to reveal structures such as homogeneous blocks. The absence of ground truth in real problems hinders the objective assessment of the components of learning algorithms, so that simulated data are particularly worthwhile. However, their design raise issues that have no equivalent in one-way clustering. Specifically, the quantification of the intrinsic difficulty of the learning problem, akin to the Bayes' error rate in clustering, is problematic, due to the dual nature of each dimension of the data table. We revisit the fundamentals of co-clustering by defining appropriate losses, and apply the founding principles of statistical learning to derive the corresponding Bayes' rules and risks. Then, we describe the difficulties related to the estimation of these quantities, which have received little attention until now in spite of their aftermath in the evaluation of learning algorithms. Finally, we exemplify the artificial data design process with latent-block models. We created a repository comprising more than 100 data tables, which is accessible online, thus providing the first publicly available series of co-clustering problems with assessed intrinsic difficulty for tables of reals, counts and binary data.

Keywords: Bayes' risk, co-clustering, latent block model, benchmarking, simulated data.

1. Introduction

Generally, the evaluation of unsupervised learning techniques on real data is difficult because there are often several legitimate views on data. For example, customers may be equally properly categorized from their purchases, residence, social status, among others. Hence, measuring the relevance of clustering by the ability to recover one or the other label is not *the* definitive apposite evaluation protocol. As a result, artificial data have been a traditional hallmark in the experimental evaluation of clustering tools. A controlled setup can truthfully test to what extent a given algorithm is able to retrieve an assumed cluster structure, by investigating its qualities and defaults in more or less favorable

cases. Once the practical strengths and limits are well understood, the behavior on real data becomes easier to foresee and interpret.

A minimalist experimental design for simulation studies in the clustering framework would consider the following ingredients:

- sample size, that is, the number of lines of the data table;
- dimensionality of the classified vectors, that is, the number of columns of the data table;
- number of clusters;
- difficulty of the task.

Then, many other configurations may be relevant for the study, according to the assumptions on the generation process. In this paper, we study the transposition of the four items mentioned above to co-clustering analysis, and we concentrate on the last bullet point.

Co-clustering aims at identifying block patterns in a data table, from a joint clustering of rows and columns. Several variants of this problem have been proposed under diverse appellations: bi-clustering, cross-clustering, two-mode partitional, or simultaneous clustering (Banerjee et al., 2007). Block-clustering organizes simultaneously the rows and columns of the data table to unveil homogeneous aligned blocks, also called checkerboard structure (Kluger et al., 2003) or grid clustering (Seldin and Tishby, 2010). This process, which identifies a pair of partitions in rows and columns, has been studied since 1965 in statistics (Good, 1965), with recent interests in various fields, ranging from graph analysis (Daudin et al., 2008), machine learning (Banerjee et al., 2007), data mining (Berkhin, 2006) and genomics (Jagalur et al., 2007).

There is yet no standard benchmark for co-clustering algorithms, as few freely available real data are amenable to an objective evaluation, due to the lack of ground truth. As a result, each author proposes his own simulation protocol for analyzing the strengths and weaknesses of learning algorithms. Reproducibility is often precluded by the sketchy description of experimental conditions and the absence of online data. Besides, quantitative assessments are also difficult to appraise due to the heterogeneity of evaluation protocols and the fact that intrinsic difficulty of the tasks is unknown to the reader.

In this paper, we first look for consensual measures of similarity between two co-clusterings. We adopt a classification perspective, where the primary objective is to recover row and column classes. In this framework, the difficulty of a co-clustering task has still to be defined appropriately. We go back to the founding principles of statistical learning by defining suitable losses, Bayes' rules and risks. Then, we describe our solutions to some of the practical problems raised by the estimation of these quantities.

We then describe a simulation protocol based on the latent-block generative model. This model assumes a strong homogeneity within blocks, whose entries only differ by random fluctuations around a common mode or mean value. Our

repository, freely available from `http://www.hds.utc.fr/coclustering`, incorporates more than 100 simulated tables of reals, counts, and binary data. It can be used for several purposes, such as the analysis of the behavior of estimation algorithms or model selection procedures. To our knowledge, this repository gathers the first publicly available series of co-clustering problems with assessed intrinsic difficulty. Furthermore, several plausible "ground truths" are provided for each data table, in order to account for the diversity of legitimate clusterings complying with the block-clustering assumptions.

2. Notations

Throughout this paper, we will use boldface lowercase for vectors, boldface uppercase for matrices, calligraphic uppercases for sets, and medium uppercase for random variables, whatever their type. The $n \times d$ data table to be processed is denoted $\mathbf{X} = (\mathbf{x}_1^\mathsf{T}, \ldots, \mathbf{x}_n^\mathsf{T})^\mathsf{T}$, with $(\mathbf{x}_i)_j = x_{ij}$, and $x_{ij} \in \mathcal{X}$ may be a real, a positive integer or a binary variable. We will systematically use i as a row index and j as a column index and, when not detailed in sums or products, i goes from 1 to n and j goes from 1 to d. Column j of $\mathbf{X}$ will be denoted $\mathbf{x}^j$, so that $\mathbf{X} = (\mathbf{x}^1, \ldots, \mathbf{x}^m)$. The row labeling in g groups, which is denoted $\mathbf{z} = (z_1, \ldots, z_n)$, takes its values in $\mathcal{Z} = \{1, \ldots, g\}^n$. Similar notations are used for the column labeling in m groups, with $\mathbf{w} \in \mathcal{W} = \{1, \ldots, m\}^d$. Probabilities will be denoted by $\mathbb{P}(\cdot)$, expectations by $\mathbb{E}[\cdot]$ and probability distributions, on either discrete or continuous variables by $p(\cdot)$.

3. Primary Definitions

Statistics and machine learning have a long tradition of quantifying the difficulty of learning problems and the achievements of learning algorithms by some objective criterion. The paramount framework for this evaluation is Bayes' decision theory, where the discrepancy between observations and predictions is measured by a loss, and where the ultimate goal of learning is to find the prediction rule with minimal risk, that is, with minimum expected loss. The difficulty of the learning problem itself may then be characterized by this Bayes' risk.

3.1. Classification Perspective

The numerous approaches to co-clustering may be split into two broad categories:

- model based, where mixture models and variants thereof use latent variables to define the row and column clusters that form block clusters (Govaert and Nadif, 2003; Shan and Banerjee, 2008; Wyse and Friel, 2010);
- reconstruction based, where the problem is formalized as a matrix approximation problem, using different sets of constraints for the approximation and dissimilarity measures for the reconstruction (Hartigan, 1972; Govaert, 1977, 1995; Banerjee et al., 2007).

The definition of relevant losses is an issue that has mainly been addressed from a reconstruction perspective, by measuring the discrepancy between the original data table and its summary provided by clustering (Hartigan, 1972; Govaert, 1995; Banerjee et al., 2007). This reconstruction-based viewpoint has the advantage of measuring goodness-of-fit from what is actually observed, that is, the data table, but it has two drawbacks: first, it is based on generalized inertia criteria, which are ancillary to the grouping of rows and columns itself; second, it requires to posit a relevant distance between matrices.

In this paper, we adopt the classification viewpoint, where losses are defined from the actual partitioning of rows and columns into sub-populations, by measuring the disagreement between the assignments of rows and columns to clusters. The main drawback of this viewpoint is to be based on unobserved quantities (the true row and column class assignments), and its main advantage resides in its ability to infer a suitable distance between rows and columns from data.

3.2. Statistical Unit

Before defining the loss incurred by a prediction-observation pair, one has to define the statistical unit of interest. In standard one-way clustering, the training set is usually represented as a data table, where each row represents an object measured on a set of variables, and each column represents a variable recorded on a set of objects. The objective of clustering is stated in terms of objects, that is, rows of the data table, which are the statistical units of interest. In co-clustering, the aim is to partition the data table into homogeneous blocks of rows×columns. These subtables belong to the Cartesian product of the sets of rows and columns, and they define a lattice on the whole table, with interdependencies between subtables, so that the relevant statistical unit is the data table itself.

3.3. Loss Functions

Considering the data table as a whole, several loss functions may be defined for measuring the discrepancy between the true label $(\mathbf{z}, \mathbf{w})$ and the decision $(\mathbf{z}', \mathbf{w}')$. Some criteria, based either on the Rand index or mutual information have been specifically tailored for clustering (Meilă, 2007), and could also be used to evaluate co-clustering. Here, we will only develop our arguments on very simple general-purpose classification losses, to focus on some peculiar issues related to co-clustering itself.

A first loss considers that the overall table assignment is either wrong or correct:

$$\ell_{\text{full}}((\mathbf{z}, \mathbf{w}), (\mathbf{z}', \mathbf{w}')) = 1 - \delta_{(\mathbf{z},\mathbf{w}),(\mathbf{z}',\mathbf{w}')} \ , \tag{1}$$

where δ is the Kronecker delta. This loss function only rewards the exact recovery of all row and column assignments, and is too stringent for evaluation purposes in the most common cases where the membership of some rows or columns is ambiguous. In these situations, we want a more graded reward when

approaching correct decisions, and a reasonable loss is the ratio of misclassified entries in the data table:

$$\ell_{\text{item}}((\mathbf{z},\mathbf{w}),(\mathbf{z}',\mathbf{w}')) = 1 - \frac{1}{nd}\sum_{i,j}\delta_{(z_i,w_j),(z'_i,w'_j)} \tag{2}$$

$$= \underbrace{\ell_{\text{row}}(\mathbf{z},\mathbf{z}')}_{1-\frac{1}{n}\sum_i \delta_{z_i,z'_i}} + \underbrace{\ell_{\text{col}}(\mathbf{w},\mathbf{w}')}_{1-\frac{1}{d}\sum_j \delta_{w_j,w'_j}} - \ell_{\text{row}}(\mathbf{z},\mathbf{z}')\ \ell_{\text{col}}(\mathbf{w},\mathbf{w}')\ , \tag{3}$$

where ℓ_{row} and ℓ_{col} are the row and the column error rate respectively. Note that this loss differs from the row and column classification error that are sometimes used for evaluating co-clustering (see for example Shan and Banerjee, 2008). This loss may incorporate weights to penalize some particular errors more heavily, and it may also serve as a basis for computing more demanding criteria such as precision, recall, area under the receiver operating characteristic, and so forth. We will not pursue in these directions where the standard tools developed for classification apply, and will instead elaborate on the two basic losses (1) and (2) in the co-clustering framework. Finally, though criteria based on the estimated probabilities $\hat{\mathbb{P}}(Z=\mathbf{z},W=\mathbf{w}|\mathbf{X})$ may also be relevant in a classification viewpoint of co-clustering, we will not elaborate on this line since computing probabilities such as $\mathbb{P}(Z=\mathbf{z},W=\mathbf{w}|\mathbf{X})$ or $\mathbb{P}(Z_i=z_i,W_j=w_j|\mathbf{X})$ is intractable (more details will be given in the following sections).

3.4. Bayes' Classifiers

The Bayes' classifier associated to the losses of Section 3.3 is a function from $\mathcal{X}^{n\times d}$ to $\mathcal{Z}\times\mathcal{W}$. For ℓ_{full}, it is defined as follows:

$$\begin{aligned}\mathbf{h}_{\text{full}}(\mathbf{X}) &= \operatorname*{argmin}_{(\mathbf{z},\mathbf{w})} \mathbb{E}\left[\ell_{\text{full}}((Z,W),(\mathbf{z},\mathbf{w}))|\mathbf{X}\right] \\ &= \operatorname*{argmax}_{(\mathbf{z},\mathbf{w})} \mathbb{P}(Z=\mathbf{z},W=\mathbf{w}|\mathbf{X})\ ,\end{aligned} \tag{4}$$

that is, the Bayes' classifier for loss ℓ_{full} (1) is the maximum a posteriori (MAP) classifier. For ℓ_{item}, the Bayes' classifier is defined likewise yielding:

$$\mathbf{h}_{\text{item}}(\mathbf{X}) = \operatorname*{argmax}_{(\mathbf{z},\mathbf{w})} \sum_{i,j}\mathbb{P}(Z_i=z_i,W_j=w_j|\mathbf{X})\ . \tag{5}$$

Note that there is no reason for the two decision rules to agree: the first one picks the most probable label among the $g^n\times m^d$ possible ones, while the second one is only sensitive to the $n\times g\times d\times m$ marginal probabilities[1]. In general, the rules will differ, a notable exception being the vacuous block-clustering problem where all conditional label variables $\{(Z_1|X),\ldots,(Z_n|X),(W_1|X),\ldots,(W_d|X)\}$ are independent.

[1] Note however that there is no direct way of computing these marginals, which should thus be evaluated by summing the entries of the original $g^n\times m^d$ table.

3.5. Bayes' Risks

The Bayes' risk is the expected loss for the Bayes' classifier. In classification, it represents the minimal expected loss incurred when classifying an arbitrary "test" example, that is, the smallest expected generalization error among all possible classification rules. In co-clustering, the Bayes' risk is transcribed likewise, by the generalization performance of the Bayes' classifier on "test" data tables.

While the generalization to arbitrary examples is typically desired when establishing a decision rule, generalizing to other tables is usually not intended in the co-clustering framework as there is commonly a single table of interest. When generalizing is mentioned, it is more customarily related to the clustering of additional rows or columns of the table under study. Hence, the usual notion of Bayes' risk is not appropriate in co-clustering, and conditioning the risk on the observed table is more sensible. In this respect, co-clustering is similar to a fixed design classification experiment that would aim at inferring the labels on a fixed collection of data points.

In what follows, we will mostly refer to the conditional risk defined from the loss ℓ_{item}:

$$r_{\text{item}}(\mathbf{z}, \mathbf{w}|\mathbf{X}) = \mathbb{E}[\ell_{\text{item}}((Z, W), (\mathbf{z}, \mathbf{w}))|\mathbf{X}] \ , \tag{6}$$

where the reference to $\mathbf{X}$ will be dropped when clear from the context. The conditional Bayes' risk will then be denoted $r_{\text{item}}(\mathbf{h}_{\text{item}}(\mathbf{X}))$.

At this point, we have to make two clarifying points. First, the parallel with fixed design does not imply that modeling data tables by a random variable is inappropriate: the use of a probabilistic model for non-repeatable events is perfectly legitimate out of an exclusively frequentist interpretation of probabilities. Second, as the inadequacy of the Bayes' risk stems from considering the data table as the statistical unit, we would like to stress that other choices would lead to further problems. As generalization usually means clustering new rows or columns of the table, one could argue that the latter are the relevant statistical units, but adding new rows or columns respectively alters the column and row clustering, because the number of rows and columns respectively define the dimension of the space where column and row clustering are performed. The duality of rows and columns and the one of examples and variables imply that a co-clustering problem is characterized by the size of the data table. Further issues related to the size of the data table will be discussed in Section 4.5.

4. Practicalities

In this section, we address the issues related to the computation of the quantities introduced in the previous section, and we discuss other topics pertaining to the difficulty of co-clustering tasks: global and row and column performances and table size. This discussion is illustrated with examples where data tables are generated from latent-block models that will be introduced later in Section 5.

4.1. Working Assumptions

The loss functions presented in Section 3.3 are bi-variate functions of pairs of row and column labels. Their minimization is hard, since they require an evaluation of all possible labels on the whole table. For data tables of size $n \times d$ considering g row and m column labels, we thus have to consider the whole discrete label domain of cardinality $g^n m^d$. This tractability issue has obviously some outcomes regarding the learning procedures that are out of the scope of the present paper, but it also impacts the evaluation of the Bayes' classifiers and Bayes' risks, which cannot be computed in polynomial time if no decomposition of $p(\mathbf{z}, \mathbf{w}|\mathbf{X})$ is available. We state below our two main working assumptions. They only partially satisfy our needs regarding the computation of the quantities introduced in the previous section, but we believe that they are mild enough to be realistic in many real applications.

The first assumption extends the usual assumption of one-way clustering of prior independence of classes to the joint labeling of rows and columns. The second one states that the conditional distribution of the data table given its row and column labels factorizes as the conditional distribution of its rows, and also as the conditional distribution of its columns. These assumptions ensure that modelling is insensitive to row and column permutations, which is a usual desideratum when there is no structural linkage between rows and columns. Furthermore, once postulated on the whole table $\mathbf{X}$, they apply to any subtable constructed from an arbitrary row and column extraction of $\mathbf{X}$.

Assumption 1. *All labels are independent:*

$$p(\mathbf{z}, \mathbf{w}) = \prod_i p(z_i) \prod_j p(w_j) \ .$$

This assumption has two notable corollaries, namely:

1. the row labels and the column labels are independent:

$$p(\mathbf{z}) = \prod_i p(z_i)$$
$$p(\mathbf{w}) = \prod_j p(w_j) \ ;$$

2. the row and column labels are also jointly independent:

$$p(\mathbf{z}, \mathbf{w}) = p(\mathbf{z})\, p(\mathbf{w}) \ .$$

The first corollary of Assumption 1 corresponds to the independence assumption of the one-way clustering of rows and columns. The second corollary regarding the mutual independence of row and column labels may be misinterpreted: we stress here that the unconditional independence of Z and W does not imply their conditional independence knowing the data, that is $p(\mathbf{z}, \mathbf{w}|\mathbf{X}) \neq p(\mathbf{z}|\mathbf{X})\, p(\mathbf{w}|\mathbf{X})$. For example, in market analysis, consumers and product segments can be considered as independent variables, but the purchase of a given product may convey some information about the buyer.

Assumption 2. *Given the column labeling, the conditional distributions of rows given their labels factorize. Correspondingly, given the row labeling, the conditional distributions of columns given their labels factorize:*

$$\begin{aligned} p(\mathbf{X}|\mathbf{z},\mathbf{w}) &= \prod_i p(\mathbf{x}_i|z_i,\mathbf{w}) \\ &= \prod_j p(\mathbf{x}^j|\mathbf{z},w_j) \ . \end{aligned}$$

Hence, co-clustering a data table amounts to clustering its rows and its columns. *Two-way* clustering then results from partial label sharing, whereby dependencies are introduced between rows and columns: all row labels $(z_i, \mathbf{w})$ share the same $\mathbf{w}$, and all column labels $(\mathbf{z}, w_j)$ share the same $\mathbf{z}$.

4.2. Computing Bayes' rules

The Bayes' classifiers are essential elements in the design of a simulation protocol, since they assess the intrinsic difficulty of a given problem. Equivalently, they represent ultimate references for the classifiers estimated by learning algorithms.

Proposition 1. *Under Assumptions 1 and 2, we have:*

$$\begin{aligned} p(\mathbf{z}|\mathbf{w},\mathbf{X}) &= \prod_i p(z_i|\mathbf{w},\mathbf{x}_i) \\ p(\mathbf{w}|\mathbf{z},\mathbf{X}) &= \prod_j p(w_j|\mathbf{z},\mathbf{x}^j) \ . \end{aligned}$$

The proofs follow from a few lines of algebra.

Proposition 1 is not sufficient for the direct computation of the MAP classifier $\mathbf{h}_{\text{full}}(\mathbf{X})$, but it allows for an iterative optimization scheme alternating the optimization with respect to row and label classes:

Algorithm 1: Approximation of $\mathbf{h}_{\text{full}}(\mathbf{X})$

input : distribution parameters $\boldsymbol{\theta}$, table $\mathbf{X}$, labels $(\mathbf{z}^0, \mathbf{w}^0)$
output: $\widetilde{\mathbf{h}}_{\text{full}}(\mathbf{X}) = (\tilde{\mathbf{z}}, \widetilde{\mathbf{w}})$
initialize $(\tilde{\mathbf{z}}, \widetilde{\mathbf{w}}) \longleftarrow (\mathbf{z}^0, \mathbf{w}^0)$, $(\mathbf{z}^0, \mathbf{w}^0) \longleftarrow (\mathbf{0}, \mathbf{0})$
while $(\tilde{\mathbf{z}}, \widetilde{\mathbf{w}}) \neq (\mathbf{z}^0, \mathbf{w}^0)$ **do**
  $(\mathbf{z}^0, \mathbf{w}^0) \longleftarrow (\tilde{\mathbf{z}}, \widetilde{\mathbf{w}})$
1 $\tilde{\mathbf{z}} \longleftarrow \text{argmax}_{\mathbf{z}}\, p(\,\mathbf{z}\,|\widetilde{\mathbf{w}}, \mathbf{X}; \boldsymbol{\theta})$
2 $\widetilde{\mathbf{w}} \longleftarrow \text{argmax}_{\mathbf{w}}\, p(\mathbf{w}|\tilde{\mathbf{z}}\,, \mathbf{X}; \boldsymbol{\theta})$

Steps 1 and 2 of the algorithm maximize $p(\mathbf{z}, \mathbf{w}|\mathbf{X})$ with respect to $\mathbf{z}$ and $\mathbf{w}$ respectively. Thanks to Proposition 1, both steps decompose as n and d maximizations with respect to g and m discrete values. The convergence towards a local minimum in finite time follows from the strict increase of $p(\mathbf{z}, \mathbf{w}|\mathbf{X})$,

which is trivially upper bounded by 1, and the finite number of configurations. In the following experiments, convergence was always reached in less than ten iterations (typically in two–three iterations), and the final solution was stable with respect to initialization in terms of complete data likelihood $p(\tilde{\mathbf{z}}, \widetilde{\mathbf{w}}, \mathbf{X})$.

The situation is more complex regarding $\mathbf{h}_{\text{item}}(\mathbf{X})$, since the maximization of $p(z_i, w_j|\mathbf{X})$ has to consider the information conveyed by the whole data table on the class of its (i, j)th element. Such interdependencies are extremely difficult to handle globally, and the assumptions that could lead to efficient computations resembling Algorithm 1 are way too crude. As a result, we will use the output of Algorithm 1, that is, $\widetilde{\mathbf{h}}_{\text{full}}(\mathbf{X})$, as a surrogate for $\mathbf{h}_{\text{item}}(\mathbf{X})$.

4.3. Computing the Conditional Risk

The conditional Bayes' risk, presented in Section 3.5 is defined by a conditional expectation, given the observed table $\mathbf{X}$. Even when the analytical expression of the joint distribution is known, computing this expectation requires an exponential time since the conditional distribution of labels $p(\mathbf{z}, \mathbf{w}|\mathbf{X})$ is characterized by a probability table of size $g^n m^d$. It could be estimated by drawing independent realizations of labels for the observed data table $(Z, W|\mathbf{X})$, but the available generative models operate the other way round, by drawing $\mathbf{X}$ from the realization $(\mathbf{z}, \mathbf{w})$ and additional distribution parameters.

A basic approximation consists in replacing the conditional expectation with respect to labels by the empirical distribution, that is, using the generated partition, say $(\mathbf{z}^0, \mathbf{w}^0)$. More refined procedures draw labels from $p(\mathbf{z}, \mathbf{w}|\mathbf{X})$ by Markov chain Monte Carlo. Our working assumptions allow for an efficient blockwise implementation of Gibbs sampling:

Proposition 2. *Let $\mathbf{z}_{\setminus i}$ denote the vector $\mathbf{z}$ deprived from its ith component, Assumptions 1 and 2 imply:*

$$p(z_i|\mathbf{z}_{\setminus i}, \mathbf{w}, \mathbf{X}) = p(z_i|\mathbf{w}, \mathbf{x}_i)$$
$$p(w_j|\mathbf{z}, \mathbf{w}_{\setminus j}, \mathbf{X}) = p(w_j|\mathbf{z}, \mathbf{x}^j) \ ,$$

The proofs follow from a few lines of algebra.

Hence, the Gibbs sampler simply iterates the generation of n and d independent categorical variables with respectively g and m outcomes. Note that there is no "label switching" problem in this process, since the sampler uses the true parameters of the distribution.

Figure 1 illustrates that the apparent error rate may be extremely variable, and that averaging errors from many Gibbs samples has then a dramatic stabilization effect on the evaluation measure. The right-hand-side histogram summarizes the differences between the apparent error rate of the MAP classifier[2] $\ell_{\text{item}}((\mathbf{z}^0, \mathbf{w}^0), \widetilde{\mathbf{h}}_{\text{full}}(\mathbf{X}))$, where the labels $(\mathbf{z}^0, \mathbf{w}^0)$ have actually been used to generate $\mathbf{X}$, and the estimation of the conditional risk $r_{\text{item}}(\widetilde{\mathbf{h}}_{\text{full}}(\mathbf{X}))$ provided

[2]More precisely, the classifier supplied by Algorithm 1.

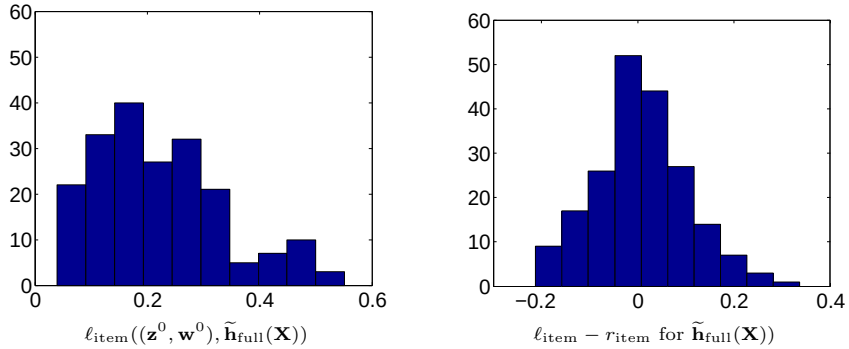

Figure 1: Apparent error rate (left) and differences between apparent error rate and conditional risk (right) for the MAP classifier on 50×50 data tables generated from the same distribution, with 3×3 clusters, and estimated (unconditional) Bayes' risk of 20%.

by averaging 2,000 Gibbs samples[3]. We see that the order of magnitude of the differences is comparable with the quantities themselves. Hence, the evaluation from the sole labels that generated a data table can be extremely noisy, and considering a large set of possible labels has then a noticeable stabilizing effect. For each data table provided in our repository, we included 2,000 such Gibbs labels for enabling accurate evaluations.

4.4. Controlling the Conditional Risk

The spread of the distribution of the apparent error rate displayed in the left-hand-side of Figure 1 is not only due to the variability with respect to labels. The histogram in the left-hand-side of Figure 2 testifies that remarkable spread in performances may still be observed for the conditional risk, which integrates error over the label distribution.

Hence, instability also arises from having a unique observed data table. Even when the distribution is fixed, there are some favorable draws, with well-separated data, and much more difficult ones, with more scatter, because the amount of available data is not large enough to be representative of the whole distribution. This phenomenon, which can also be observed in simple one-way clustering when the evaluation is based solely on the available sample, is alleviated as the table size grows, as shown in the right-hand-side of Figure 2, where the size of the error bars decreases with the table size.

For our purpose, these observations imply that the probability distribution of data tables is not specific enough to characterize their intrinsic co-clustering difficulty. We thus have to assess the conditional risk, at the level of each table. Our simulation protocol relies on a form of rejection sampling strategy that enables a precise control of the difficulty of each data in our repository.

[3]The standard error of this estimation was empirically estimated to be below 1%.

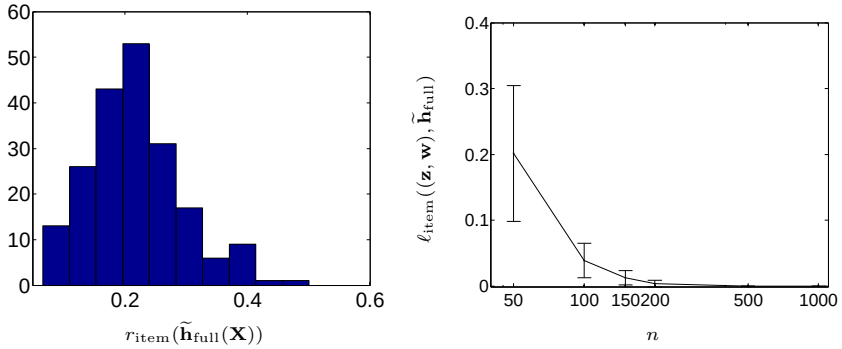

Figure 2: Conditional risk computed for square data tables ($m = n$) with 3×3 clusters whose entries are generated from the same distribution: left, histogram computed from 50×50 tables; right, average loss and interquartile range *vs.* table size.

4.5. Table Size and Separability

Co-clustering is subject to an unusual phenomenon that is clearly on view in the right-hand-side of Figure 2: for a given distribution on the entries of the table, the expected loss decreases as the table size grows. We stress that there is no learning process in the curve displayed; the loss is computed for the MAP decision rule, using the knowledge of the true distribution. Hence, the error decrease reflects a property of the table distribution: its size plays an important role in setting the Bayes' risk, that is, the intrinsic difficulty of the task. This is in contrast with most learning scenarios, where more data usually leads to better estimation performance, but does not impact the Bayes' risk.

To understand this phenomenon, consider the representation of a $n \times d$ table as n d-dimensional vectors. When the distribution of the vector entries differ (that is, when the row clusters differ), the overall dissimilarity between vectors will grow as d grows. Figure 3 displays the principal components of such vectors, extracted from two tables with $d = 50$ in the left plot and $d = 500$ in the right plot. The distribution describing the classes and their probabilities is identical in both plots, but while the clusters in dimension $d = 50$ are highly overlapping, they are well separated in $d = 500$.

Figures 2 and 3 illustrate a systematic but little known property of block-clustering: the distribution of the table entries being fixed, the Bayes' risk decreases with the table size. More formal arguments, already developed for the more constrained stochastic block model (Celisse et al., 2011), could be transposed to co-clustering. Intuitively, this decrease can be understood by considering that the table enlargement in one dimension results in more redundancy in the other dimension.

4.6. Co-Clustering Structure

From an optimization viewpoint, the losses defined in Section 3.3 are objective functions to be minimized with respect to $(\mathbf{z}', \mathbf{w}')$. This can be rephrased

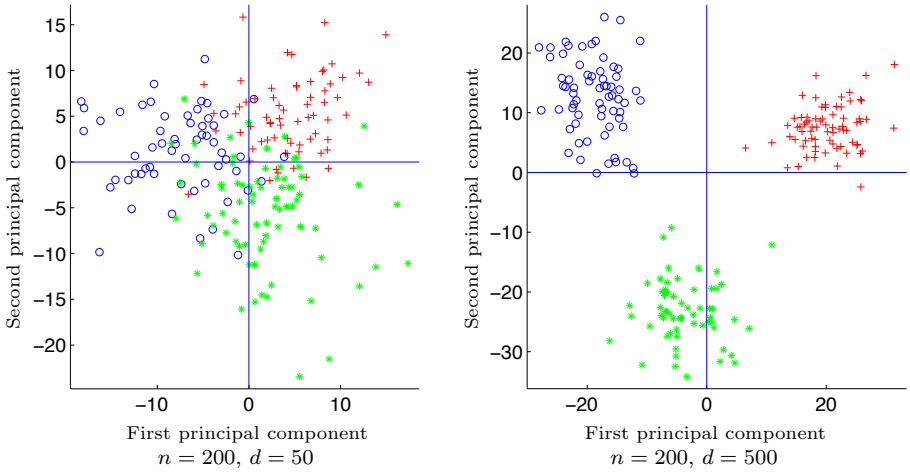

Figure 3: Projections of the rows of two data tables on the two first column eigenvectors. The distribution of table entries is identical, but table sizes differ, with $n = 200$ rows and $d = 50$ (left) or $d = 500$ (right).

as two nested clustering problems, for example through the following decomposition:

$$\min_{(\mathbf{z}',\mathbf{w}')\in\mathcal{Z}\times\mathcal{W}} \ell((\mathbf{z},\mathbf{w}),(\mathbf{z}',\mathbf{w}')) = \min_{\mathbf{z}'\in\mathcal{Z}} J_{\mathbf{z},\mathbf{w}}(\mathbf{z}') \ ,$$
$$\text{where } \ J_{\mathbf{z},\mathbf{w}}(\mathbf{z}') = \min_{\mathbf{w}'\in\mathcal{W}} \ell((\mathbf{z},\mathbf{w}),(\mathbf{z}',\mathbf{w}')) \ .$$

The inner optimization problem of evaluating $J_{\mathbf{z},\mathbf{w}}$ is a clustering problem with respect to columns, and the outer problem of minimizing $J_{\mathbf{z},\mathbf{w}}$ is a clustering problem with respect to rows. If the joint optimization problem is separable in $\mathbf{z}'$ and $\mathbf{w}'$, it is not a genuine co-clustering problem, but rather a pair of clustering problems. This happens in particular if clustering is trivial with respect to one dimension, in which case co-clustering boils down to standard one-way clustering with respect to the other dimension. Thus, to ensure the generation of tables with undisputed co-clustering structure, the row and column losses defined in (3) should ideally be of the same order of magnitude. This goal can be achieved either by setting appropriate distribution parameters or by modifying the relative dimensions of the table.

5. Simulation Protocol

The objective evaluation measures discussed until now require the knowledge of the true labels, which is rarely available for real data. This section describes the latent-block model (Govaert and Nadif, 2003), which has been used to generate the benchmark data of our repository for tables of reals, counts and binary

entries. Then, we introduce a simulation protocol that provides controls on the difficulty of the tasks, as demonstrated by the examples provided in the repository.

5.1. Latent-Block Model

The latent-block model is a probabilistic generative model that generalizes mixture models (Govaert and Nadif, 2003). With regard to the components of the distribution, that is, the conditional distributions of the observed data table given the unobserved row and column labels, the model postulates that the table entries are independent given the row and column labels:

$$p(\mathbf{X}|\mathbf{z},\mathbf{w}) = \prod_{i,j} p(x_{ij}|\mathbf{z},\mathbf{w}) \ .$$

Also, conditionally on their row and column labels, all table entries are assumed to be generated independently from the other labels:

$$\forall (i,j) \in \{1,\dots n\} \times \{1,\dots,d\}\,, \ \ p(x_{ij}|\mathbf{z},\mathbf{w}) = p(x_{ij}|z_i,w_j) \ ,$$

and the within-block entries are identically distributed. As a result, the model complies with Assumption 2, and the conditional distribution of the data table given labels is written as follows:

$$p(\mathbf{X}|\mathbf{z},\mathbf{w};\boldsymbol{\alpha}) = \prod_{i,j} p(x_{ij}|z_i,w_j;\boldsymbol{\alpha}_{z_i w_j}) \ ,$$

where $\boldsymbol{\alpha}_{z_i w_j}$ is the parameter of $p(x_{ij}|z_i,w_j)$, which is only indexed by the row and columns labels (z_i,w_j), rendering that all entries belonging to the same block are identically distributed.

The unconditional distribution of the table is then defined by the mixture distribution of all components. This mixture complies with Assumption 1 by supposing the independence of labels, which are furthermore identically distributed, leading to the following decomposition:

$$p(\mathbf{X};\boldsymbol{\theta}) = \sum_{(\mathbf{z},\mathbf{w})\in\mathcal{Z}\times\mathcal{W}} p(\mathbf{z};\boldsymbol{\pi})\ p(\mathbf{w};\boldsymbol{\rho}) \prod_{i,j} p(x_{ij}|z_i,w_j;\boldsymbol{\alpha}_{z_i w_j}) \ , \tag{7}$$

where $\boldsymbol{\pi} = (\pi_1,\dots \pi_g)$ is the probability of row labels, $\boldsymbol{\rho} = (\rho_1,\dots,\rho_m)$ is the probability of column labels, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_{z_1 w_1},\dots,\boldsymbol{\alpha}_{z_g w_m})$ is the set of parameters describing the component distributions, and $\boldsymbol{\theta} = (\boldsymbol{\pi},\boldsymbol{\rho},\boldsymbol{\alpha},n,d)$ is a shortcut notation for all parameters. While the number of row and column classes is implicit in $\boldsymbol{\theta}$ through $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$, the table size, which appears in (7) through the definition of $(\mathcal{Z},\mathcal{W})$ has to be included for completeness. Though not enforced by the model, the form of $p(x_{ij}|z_i,w_j;\boldsymbol{\alpha}_{z_i w_j})$ is typically identical for each block, such as Gaussian for data tables of reals (Govaert and Nadif, 2003), Poisson for contingency tables (Govaert and Nadif, 2007), or Bernoulli for binary data (Govaert and Nadif, 2008).

5.2. Simulation Details

Our simulation protocol consists in four steps. In the first step, one chooses the type of simulated data table (real, count, or binary), which will result in selecting an adequate distribution for the table entries (respectively Gaussian, Poisson, or Bernoulli). Additionally, one chooses the size of the table $n \times d$, the number of row and column clusters $g \times m$, and the difficulty of the task, by setting the conditional risk $r_{\text{item}}(\widetilde{\mathbf{h}}_{\text{full}}(\mathbf{X}))$.

In the second step, the distribution parameters $\boldsymbol{\pi}$, $\boldsymbol{\rho}$, and $\boldsymbol{\alpha}$ are generated, with slight modifications according to the type of data. Some options allow to limit the flexibility of the model, for example by considering identical class probabilities or dispersion parameters across blocks. The $g \times m$ means or modes of the block entries are roughly symmetrically located in order to have similar overlaps throughout the data table in rows and columns. For Gaussian and Bernoulli distributions, the dispersion parameters (that is, variances and deviations from the mode) are randomly chosen with the same order of magnitude.

In the third step, the parameters $\boldsymbol{\alpha}$ are scaled to calibrate the Bayes' risk at the prescribed conditional risk value. A series of 200 data tables is generated by first drawing the partitions $\mathbf{z}^0$ and $\mathbf{w}^0$ from multinomial distributions of parameters $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$ respectively. Then, the entries of the table are generated by independent draws from $p(x_{ij}|z_i, w_j; \boldsymbol{\alpha}_{z_i w_j})$ and the apparent error rate $\ell_{\text{item}}((\mathbf{z}^0, \mathbf{w}^0), \widetilde{\mathbf{h}}_{\text{full}}(\mathbf{X}))$ is computed. The parameters $\boldsymbol{\alpha}$ are updated by setting a scale parameter by trial and error (dichotomy) as long as the apparent error rate is not centered around the stipulated conditional risk, and the generation process is repeated.

In the last step, with the table distribution parameters well calibrated, one or more data tables complying with the specified conditional risk are supplied. A table is generated as above, and its conditional risk is computed by Gibbs sampling. If the latter does not match the prescribed conditional risk, the generation and evaluation process is repeated; otherwise, the table, its series of Gibbs labels and the distribution parameters are recorded in the repository. Rejection sampling can be severe (70%) for small tables, but the rejection rate decreases with table size.

Acknowledgments

We gratefully acknowledge the partial support from the French National Research Agency (ANR) under grant ClasSel ANR-08-EMER-002, the PASCAL2 Network of Excellence, and the European ICT FP7 under grant No 247022 - MASH.

6. Conclusion

The co-clustering literature, in spite its forty years long history, has not yet come out with a consensual way of benchmarking algorithms. When adopting a classification view of co-clustering, the primary objective is to recover the row

and column partitions. The objective assessment of the intrinsic difficulty of a co-clustering problem is then complicated, due to the absence of supervision, and because co-clustering considers data tables as statistical units, which is atypical in statistical learning. Regarding the first point, simulated data are particularly worthwhile, but we surveyed a number of hurdles on the road of the controlled design of artificial data tables, owing to the dual nature of each dimension of the data table, that has no equivalent in one-way clustering.

We began this study by arguing that the relevant statistical should be the data table. Then, we proposed suitable losses, Bayes' rules and risks, and described the difficulties related to the estimation of these quantities. The losses we consider are simple, to avoid distraction from the important specificities of co-clustering, such as the hardness of computing the optimal partitions for a fully known model.

Then, we showed that, other parameters of the distribution being equal, the table size had important outcomes regarding Bayes' error rates. This size is thus an important feature of table distributions, which is not described by the distribution of their entries.

Finally, we exemplified the artificial data design process with latent-block models. Our simulation protocol allows to produce co-clustering problems whose controlled difficulty is valuable in the analysis of the behavior of the algorithms pertaining to co-clustering. Our repository, available at `http://www.hds.utc.fr/coclustering`, gathers the first publicly available series of co-clustering problems with assessed intrinsic difficulty.

In addition to the proposal of classification-based evaluation measures and the simulation protocol, this paper exhibits two interesting phenomena whose interest goes beyond the simulation setup, in particular in regard to the evaluation of learning algorithms. First, we illustrated that, for small data tables, having a single ground truth labelling may not sufficient, because many diverse sets of labels are about equally probable for the observed data table. The histograms provided in Section 4.4 show that ignoring this diversity can result in extremely noisy evaluations. This phenomenon, which also impairs clustering to a much lesser extent, has outcomes at different levels. For practitioners, it means that several truths are possible: a series of plausible results should be displayed to the end-user. For evaluators of learning algorithms, it means that the true partition is not likely to be optimal, whatever the measure: performances based on this unique partition are not gold standards.

Second, we uncover a phenomenon already known and studied for unipartite graph models such as the stochastic block model (Celisse et al., 2011): if the number of clusters is fixed, the asymptotic Bayes' error of co-clustering goes to zero as the data table dimensions go to infinity. This theoretical analysis has yet to be transposed to the co-clustering setup, in particular to determine the growth rates of the number of rows and columns that will drive Bayes' error to zero. If our conjecture is correct, it implies that the row and column distributions converge to non-overlapping densities, which in turns implies that co-clustering boils down to a pair of independent row and column clusterings. Hence, to reveal structures that cannot be extracted by one-way clustering, the

joint partitioning of large data tables should consider a large number of clusters.

References

Banerjee, A., Dhillon, I., Ghosh, J., and Merugu, S. (2007). A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, 8:1919–1986.

Berkhin, P. (2006). A survey of clustering data mining techniques. In Kogan, J., Nicholas, C., and Teboulle, M., editors, *Grouping Multidimensional Data*, pages 25–71. Springer.

Celisse, A., Daudin, J.-J., and Pierre, L. (2011). Consistency of maximum-likelihood and variational estimators in the stochastic block model. arXiv:1105.3288v1.

Daudin, J.-J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183.

Good, I. J. (1965). *Categorization of classification*. Mathematics and Computer Science in Biology and Medicine. Her Majesty's Stationery Office, London.

Govaert, G. (1977). Algorithme de classification d'un tableau de contingence. In *First international symposium on data analysis and informatics*, pages 487–500, Versailles. INRIA.

Govaert, G. (1995). Simultaneous clustering of rows and columns. *Control and Cybernetics*, 24(4):437–458.

Govaert, G. and Nadif, M. (2003). Clustering with block mixture models. *Pattern Recognition*, 36:463–473.

Govaert, G. and Nadif, M. (2007). Clustering of contingency table and mixture model. *European Journal of Operational Research*, 183:1055–1066.

Govaert, G. and Nadif, M. (2008). Block clustering with Bernoulli mixture models: Comparison of different approaches. *Computational Statistics and Data Analysis*, 52:3233–3245.

Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67:123–129.

Jagalur, M., Pal, C., Learned-Miller, E., Zoeller, R. T., and Kulp, D. (2007). Analyzing in situ gene expression in the mouse brain with image registration, feature extraction and block clustering. *BMC Bioinformatics*, 8(Suppl 10):S5.

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

Meilă, M. (2007). Comparing clusterings–an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Seldin, Y. and Tishby, N. (2010). PAC-Bayesian analysis of co-clustering and beyond. *Journal of Machine Learning Research*, 11:3595–3646.

Shan, H. and Banerjee, A. (2008). Bayesian co-clustering. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08*, pages 530–539.

Wyse, J. and Friel, N. (2010). Block clustering with collapsed latent block models. *Statistics and Computing*, 22(2):415–428.

Appendix A. Proofs of Propositions

All propositions concern similar rows and column properties. The proofs are provided here with regard to the row property since the proof proceeds likewise for the column one. We first state a useful lemma whereby Propositions 1 and 2 can be derived in a few lines.

Lemma 1. *Under Assumptions 1 and 2, conditionally on the column labels* $\mathbf{w}$*, the variables* $(\mathbf{x}_1, z_1), \ldots, (\mathbf{x}_n, z_n)$ *are independent. Likewise, conditionally on the row labels* $\mathbf{z}$*, the variables* $(\mathbf{x}^1, w_1), \ldots, (\mathbf{x}^d, w_d)$ *are independent.*

$$p(\mathbf{X}, \mathbf{z}|\mathbf{w}) = \prod_i p(\mathbf{x}_i, z_i|\mathbf{w})$$
$$p(\mathbf{X}, \mathbf{w}|\mathbf{z}) = \prod_j p(\mathbf{x}^j, w_j|\mathbf{z}) \ .$$

Proof.

$$\begin{aligned} p(\mathbf{X}, \mathbf{z}|\mathbf{w}) &= p(\mathbf{X}|\mathbf{z}, \mathbf{w}) p(\mathbf{z}|\mathbf{w}) \\ &= \prod_i p(\mathbf{x}_i|z_i, \mathbf{w}) p(\mathbf{z}|\mathbf{w}) && \text{(Assumption 2)} \\ &= \prod_i p(\mathbf{x}_i|z_i, \mathbf{w}) \prod_i p(z_i|\mathbf{w}) && \text{(Assumption 1)} \\ &= \prod_i p(\mathbf{x}_i, z_i|\mathbf{w}) \end{aligned}$$

□

Note that Lemma 1 implies the conditional independence of rows and columns:

$$p(\mathbf{X}|\mathbf{w}) = \prod_i p(\mathbf{x}_i|\mathbf{w})$$
$$p(\mathbf{X}|\mathbf{z}) = \prod_j p(\mathbf{x}^j|\mathbf{z}) \ .$$

Appendix A.1. Proposition 1

Proposition 1 states that, under Assumptions 1 and 2, we have:

$$p(\mathbf{z}|\mathbf{w}, \mathbf{X}) = \prod_i p(z_i|\mathbf{w}, \mathbf{x}_i)$$
$$p(\mathbf{w}|\mathbf{z}, \mathbf{X}) = \prod_j p(w_j|\mathbf{z}, \mathbf{x}^j) \ .$$

Proof.

$$\begin{aligned} p(\mathbf{z}|\mathbf{w},\mathbf{X}) &= \frac{p(\mathbf{X},\mathbf{z}|\mathbf{w})}{p(\mathbf{X}|\mathbf{w})} \\ &= \frac{\prod_i p(\mathbf{x}_i, z_i|\mathbf{w})}{\prod_i p(\mathbf{x}_i|\mathbf{w})} && \text{(Lemma 1)} \\ &= \prod_i p(z_i|\mathbf{w},\mathbf{x}_i) \end{aligned}$$

□

Appendix A.2. Proposition 2

Proposition 2 states that, under Assumptions 1 and 2, we have:

$$\begin{aligned} p(z_i|\mathbf{z}_{\setminus i},\mathbf{w},\mathbf{X}) &= p(z_i|\mathbf{w},\mathbf{x}_i) \\ p(w_j|\mathbf{z},\mathbf{w}_{\setminus j},\mathbf{X}) &= p(w_j|\mathbf{z},\mathbf{x}^j) \ , \end{aligned}$$

Proof. Let $\mathbf{X}_{\setminus i}$ denote the subtable of $\mathbf{X}$ formed by all rows except row i:

$$\begin{aligned} p(z_i|\mathbf{z}_{\setminus i},\mathbf{w},\mathbf{X}) &= p(z_i|\mathbf{z}_{\setminus i},\mathbf{w},\mathbf{x}_i,\mathbf{X}_{\setminus i}) \\ &= \frac{p(\mathbf{x}_i,z_i|\mathbf{X}_{\setminus i},\mathbf{z}_{\setminus i},\mathbf{w})}{p(\mathbf{x}_i|\mathbf{X}_{\setminus i},\mathbf{z}_{\setminus i},\mathbf{w})} \\ &= \frac{p(\mathbf{x}_i,z_i|\mathbf{w})}{p(\mathbf{x}_i|\mathbf{w})} && \text{(Lemma 1)} \\ &= p(z_i|\mathbf{x}_i,\mathbf{w}) \end{aligned}$$

□