Co-clustering algorithms for histogram data
Algoritmi di Co-clustering per dati ad istogramma

Francisco de A.T. De Carvalho and Antonio Balzanella and Antonio Irpino and Rosanna Verde

Abstract One of the current big-data age requirements is the need of representing groups of data by summaries allowing the minimum loss of information as possible. Recently, histograms have been used for summarizing numerical variables keeping more information about the data generation process than characteristic values such as the mean, the standard deviation, or quantiles. We propose two co-clustering algorithms for histogram data based on the double k-means algorithm. The first proposed algorithm, named "distributional double Kmeans (DDK)", is an extension of double Kmeans (DK) proposed to manage usual quantitative data, to histogram data. The second algorithm, named adaptive distributional double Kmeans (ADDK), is an extension of DDK with automated variable weighting allowing co-clustering and feature selection simultaneously.

Abstract *Una delle principali esigenze nell'era dei big-data è quella di rappresentare gruppi di dati attraverso strumenti di sintesi che minimizzano la perdita di informazione. Negli ultimi anni, uno degli strumenti maggiormente utilizzati a tal scopo è l'istogramma. Esso fornisce una sintesi della distribuzione che genera i dati risultando più informativo delle classiche sintesi quali la media, la deviazione standard, o i quantili. Nel presente articolo, si propongono due algoritmi di co-clustering per dati rappresentati da istogrammi che estendono il classico double k-means algorithm (DK). La prima proposta chiamata "distributional double Kmeans (DDK)", è un'estensione dell'algoritmo DK a dati ad istogramma. La seconda pro-*

Francisco de A.T. De Carvalho
Centro de Informatica, Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes s/n - Cidade Universitaria, CEP 50740-560, Recife-PE, Brazil, e-mail: fatc@cin.ufpe.br

Antonio Balzanella
Università della Campania L. Vanvitelli, e-mail: antonio.balzanella@unicampania.it

Antonio Irpino
Università della Campania L. Vanvitelli, e-mail: antonio.irpino@unicampania.it

Rosanna Verde
Università della Campania L. Vanvitelli e-mail: rosanna.verde@unicampania.it

posta, chiamata adaptive distributional double Kmeans (ADDK), è un'estensione dell'algoritmo DDK che effettua la ponderazione automatica delle variabili consentendo di effettuare simultaneamente il co-clustering e la selezione delle variabili.

Key words: Co-clustering, Histogram data

1 Introduction

Histogram data are becoming very common in several applicative fields. For example, in order to preserve the individuals' privacy, data about groups of customers transactions are released after being aggregated; in wireless sensor networks, where the energy limitations constraint the communication of data, the use of suitable synthesis of sensed data are necessary; official statistical institutes collect data about territorial units or administrations and release them as histograms.

Among the exploratory tools for the analysis of histogram data, this paper focuses on co-clustering, also known as bi-clustering or block clustering. The aim is to cluster simultaneously objects and variables of a data set [1, 2, 5, 6].

By performing permutations of rows and columns, the co-clustering algorithms aim to reorganize the initial data matrix into homogeneous blocks. These blocks also called co-clusters can therefore be defined as subsets of the data matrix characterized by a set of observations and a set of features whose elements are similar. They resume the initial data matrix into a much smaller matrix representing homogeneous blocks or co-clusters of similar objects and variables. Refs. [4] presents other types of co-clustering approaches.

This paper proposes at first the DDK (Distributional Double K-means) algorithm whose aim is to cluster, simultaneously, objects and variables on distributional-valued data sets. Then, it introduces the ADDK (Adaptive Distributional Double K-means) algorithm which takes into account the relevance of the variables in the co-clustering optimization criterion.

Conventional co-clustering methods do not take into account the relevance of the variables, i.e., these methods consider that all variables are equally important to the co-clustering task, however, in most applications some variables may be irrelevant and, among the relevant ones, some may be more or less relevant than others.

ADDK and DDK use, respectively, suitable adaptive and non-adaptive Wasserstein distances aiming to compare distributional-valued data during the co-clustering task.

2 Co-clustering algorithms for distributional-valued data

Let $E = \{e_1, \ldots, e_N\}$ be a set of N objects described by a set of P distributional-valued variables denoted by $Y_j\,(1 \leq j \leq P)$. Let $\mathscr{Y} = \{Y_1, \ldots, Y_P\}$ be the set of P

distributional-valued variables and let

$$\mathbf{Y} = \left(y_{ij}\right)_{\substack{1\leq i\leq N\\ 1\leq j\leq P}}$$

be a distributional-valued data matrix of size $N \times P$ where the distributional data observed on the Y_j variable for the i-th object is denoted with y_{ij}. Our aim consists in obtaining a co-clustering of $\mathbf{Y}$, i.e, in obtaining simultaneously a partition $\mathscr{P} = \{\mathscr{P}_1, \ldots, \mathscr{P}_C\}$ of the set of N objects into C clusters and a partition $\mathscr{Q} = \{\mathscr{Q}_1, \ldots, \mathscr{Q}_H\}$ of the set of P distributional-valued variables into H clusters.

The co-clustering can be formulated as the search for a good matrix approximation of the original distributional-valued data matrix $\mathbf{Y}$ by a $C \times H$ matrix

$$\mathbf{G} = \left(g_{kh}\right)_{\substack{1\leq k\leq C\\ 1\leq h\leq H}}$$

which can be viewed as a summary of the distributional-valued data matrix $\mathbf{Y}$ (see, for example, Refs. [2, 5, 6]).

Each element g_{kh} of $\mathbf{G}$ is also called a prototype of the co-cluster

$$\mathbf{Y}_{kh} = \left(y_{kj}\right)_{\substack{e_i\in\mathscr{P}_k\\ Y_j\in\mathscr{Q}_h}}$$

Moreover, each g_{kh} is a distributional data, with a distribution function G_{kh} and a quantile function $Q_{g_{kh}}$

In order to obtain a co-clustering that is faithfully representative of the distributional-valued data set $\mathbf{Y}$, the matrix $\mathbf{G}$ of prototypes, the partition $\mathscr{P}$ of the objects and the partition $\mathscr{Q}$ of the distributional-valued variables are obtained iteratively by means of the minimization of an error function J_{DDK}, computed as follows:

$$J_{DDK}(\mathbf{G}, \mathscr{P}, \mathscr{Q}) = \sum_{k=1}^{C}\sum_{h=1}^{H}\sum_{e_i\in\mathscr{P}_k}\sum_{Y_j\in\mathscr{Q}_h} d_W^2(y_{ij}, g_{kh}) \tag{1}$$

where the d_W^2 function is the non-adaptive (squared) L_2 Wasserstein distance computed between the element y_{ij} of the distributional data matrix $\mathbf{Y}$ and the prototype g_{kh} of co-cluster $\mathbf{Y}_{kh}$

In order to propose an adaptive version of the DDK algorithm which evaluates the relevance of each variable in the co-clustering process, we still propose the local minimization of the following criterion function:

$$J_{ADDK}(\mathbf{G}, \Lambda, \mathscr{P}, \mathscr{Q}) = \sum_{k=1}^{C}\sum_{h=1}^{H}\sum_{e_i\in\mathscr{P}_k}\sum_{Y_j\in\mathscr{Q}_h} \lambda_j d_W^2(y_{ij}, g_{kh}) \tag{2}$$

where $\Lambda = (\lambda_j)_{j=1,\ldots,P}$ (with $\lambda_j > 0$ and $\prod_{j=1}^{P}\lambda_j = 1$) are positive weights measuring the importance of each distributional-valued variable.

The minimization of the J_{DDK} criterion, is performed iteratively in three steps: representation, objects assignments and distributional-valued variables assignments.

The representation step gives the optimal solution for the computation of the representatives (prototypes) of the co-clusters. The objects assignments step provides the optimal solution for the partition of the objects. Finally, the variables assignments step provides the optimal solution for the partition of the variables. The three steps are iterated until the convergence to a stable optimal solution.

The minimization of the J_{ADDK} criterion, requires a further weighting step which provides optimal solutions for the computation of the relevance weights for the distributional-valued variables.

Representation step in DDK and ADDK

In the representation step, DDK algorithm aims to find, for $k = 1,\ldots,C$ and for $h = 1,\ldots,H$, the prototype g_{kh} such that $\sum_{e_i \in \mathscr{P}_k} \sum_{Y_j \in \mathscr{Q}_h} d_W^2(y_{ij}, g_{kh})$ is minimal.

According to [3], the quantile function associated with the corresponding probability density function (*pdf*) $g_{kh}\,(1 \leq k \leq C; 1 \leq h \leq H)$ is:

$$Q_{g_{kh}} = \frac{\sum_{e_i \in \mathscr{P}_k} \sum_{Y_j \in \mathscr{Q}_h} Q_{ij}}{n_k n_h} \tag{3}$$

where n_k is the cardinality of $\mathscr{P}_k$ and n_h is the cardinality of $\mathscr{Q}_h$.

The ADDK algorithm aims to find, for $k = 1,\ldots,C$ and for $h = 1,\ldots,H$, prototype g_{kh} such that $\sum_{e_i \in \mathscr{P}_k} \sum_{Y_j \in \mathscr{Q}_h} \lambda_j d_W^2(y_{ij}, g_{kh})$ is minimal.

Under the constraints $\prod_{j=1}^{P} \lambda_j = 1$, $\lambda_j > 0$, the quantile function associated with the corresponding probability density function (*pdf*) $g_{kh}\,(1 \leq k \leq C; 1 \leq h \leq H)$ is computed as follows:

$$Q_{g_{kh}} = \frac{\sum_{e_i \in \mathscr{P}_k} \sum_{Y_j \in \mathscr{Q}_h} \lambda_j Q_{ij}}{n_k \sum_{Y_j \in \mathscr{Q}_h} \lambda_j} \tag{4}$$

Objects assignment step in DDK and ADDK

During the object assignment step of DDK, the matrix of co-cluster prototypes $\mathbf{G}$ and the partition of the distributional-valued variables $\mathscr{Q}$ are kept fixed. The error function J_{DDK} is minimized with respect to the partition $\mathscr{P}$ of objects and each object $e_i \in E$ is assigned to its nearest co-cluster prototype.

Proposition 1. *The error function J_{DDK} (Eq. 1) is minimized with respect to the partition $\mathscr{P}$ of objects when the clusters $P_k\,(k = 1,\ldots,C)$ are updated according to the following assignment function:*

$$P_k = \left\{ e_i \in E : \sum_{h=1}^{H} \sum_{Y_j \in \mathscr{Q}_h} d_W^2(y_{ij}, g_{kh}) = \min_{z=1}^{C} \sum_{h=1}^{H} \sum_{Y_j \in \mathscr{Q}_h} d_W^2(y_{ij}, g_{zh}) \right\}$$

The error function J_{ADDK} is minimized with respect to the partition $\mathscr{P}$ and each individual $e_i \in E$ is assigned to its nearest co-cluster prototype.

Proposition 2. *The error function J_{ADDK} (Eq. 2) is minimized with respect to the partition $\mathscr{P}$ of objects when the clusters $P_k\,(k = 1,\ldots,C)$ are updated according to the following assignment function:*

$$P_k = \left\{ e_i \in E : \sum_{h=1}^{H} \sum_{Y_j \in \mathscr{Q}_h} \lambda_j d_W^2(y_{ij}, g_{kh}) = \min_{z=1}^{C} \sum_{h=1}^{H} \sum_{Y_j \in \mathscr{Q}_h} \lambda_j d_W^2(y_{ij}, g_{zh}) \right\}$$

Variables assignment step in DDK and ADDK

During the variables assignment step of DDK, the matrix of prototypes $\mathbf{G}$ and the partition of the objects $\mathscr{P}$ are kept fixed. The error function J_{DDK} is minimized with respect to the partition $\mathscr{Q}$ of the distributional-valued variables and each variable $Y_j \in \mathscr{Y}$ is assigned to its nearest co-cluster prototype.

Proposition 3. *The error function J_{DDK} (Eq. 1) is minimized with respect to the partition $\mathscr{Q}$ of the distributional-valued variables when the clusters $Q_h\,(h = 1,\ldots,H)$ are updated according to the following assignment function:*

$$Q_h = \left\{ Y_j \in \mathscr{Y} : \sum_{k=1}^{C} \sum_{e_i \in \mathscr{P}_k} d_W^2(y_{ij}, g_{kh}) = \min_{z=1}^{H} \sum_{k=1}^{C} \sum_{e_i \in \mathscr{P}_k} d_W^2(y_{ij}, g_{kz}) \right\}$$

where $d_W^2(y_{ij}, g_{kz})$ is the squared L^2 Wasserstein distance.

Proposition 4. *The error function J_{ADDK} (Eq. 2) is minimized with respect to the partition $\mathscr{Q}$ of the distributional-valued variables when the clusters $Q_h\,(h = 1,\ldots,H)$ are updated according to the following assignment function:*

$$Q_h = \left\{ Y_j \in \mathscr{Y} : \sum_{k=1}^{C} \sum_{e_i \in \mathscr{P}_k} \lambda_j d_W^2(y_{ij}, g_{kh}) = \min_{z=1}^{H} \sum_{k=1}^{C} \sum_{e_i \in \mathscr{P}_k} \lambda_j d_W^2(y_{ij}, g_{kz}) \right\}$$

Weighting step for ADDK

We provide an optimal solution for the computation of the relevance weight of each distributional-valued variable during the weighting step of the ADDK algorithm.

During the weighting step of ADDK, the matrix of prototype vectors $\mathbf{G}$, the partition $\mathscr{P}$ of the objects and the partition $\mathscr{Q}$ of the distributional-valued variables are kept fixed. The error function J_{ADDK} is minimized with respect to the weights λ_j.

Proposition 5. *The relevance weights are computed according to the adaptive squared L_2 Wasserstein distance:*

If we assume that $\prod_{j=1}^{P} \lambda_j = 1, \lambda_j > 0$*, the P relevance weights are computed as follows:*

$$\lambda_j = \frac{\left\{ \prod_{r=1}^{P} \left(\sum_{k=1}^{C} \sum_{h=1}^{H} \sum_{e_i \in \mathscr{P}_k} \sum_{Y_r \in \mathscr{Q}_h} d_W^2(y_{ir}, g_{kh}) \right) \right\}^{\frac{1}{P}}}{\sum_{k=1}^{C} \sum_{h=1}^{H} \sum_{e_i \in \mathscr{P}_k} \sum_{Y_j \in \mathscr{Q}_h} d_W^2(y_{ij}, g_{kh})} \tag{5}$$

3 Conclusions

In this paper we have introduced two algorithms, based on the double k-means, for performing the co-clustering of a distributional valued data matrix. The main difference between the two algorithms is that ADDK integrates in the optimization criterion the search for a set of weights which measure the relevance of each variable. In order to evaluate the effectiveness of the proposal, we have made some preliminary test on real and simulated data with encouraging results.

References

1. Govaert G.: Simultaneous clustering of rows and columns. In: Control and Cybernetics **24** pp. 437–458 (1995)
2. Govaert G., Nadif M.: Co-Clustering: Models, Algorithms and Applications. Wiley, New York (2015)
3. Irpino, A. and Verde, R. Basic statistics for distributional symbolic variables: a new metric-based approach. In: Advances in Data Analysis and Classification,92, pp. 143–175 Springer Berlin Heidelberg (2015)
4. Pontes R., Giraldez R., Aguilar-Ruiz J.S.: Biclustering on expression data: A review. In: Journal of Biomedical Informatics, 57, pp. 163–180, (2015)
5. Rocci R., Vichi M.: Two-mode multi-partitioning. In: Computational Statistics & Data Analysis, 52 pp.1984–2003 (2008)
6. Vichi M.: Double k-means Clustering for Simultaneous Classification of Objects and Variables. In: Advances in Classification and Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization, pp. 43–52, Springer (2001)