Sparse Tensor Co-Clustering as a Tool for Document Categorization

Rafika Boutalbi[1,2]
[1]LIPADE - University of Paris
[2]TRINOV, 196 rue Saint Honoré, 75001 Paris
rafika.boutalbi@parisdescartes.fr

Lazhar Labiod[1]
[1]LIPADE - University of Paris
lazhar.labiod@parisdescartes.fr

Mohamed Nadif[1]
[1]LIPADE - University of Paris
mohamed.nadif@parisdescartes.fr

ABSTRACT

To deal with document clustering, we usually rely on document-term matrices. However, from additional available information like keywords, co-authors, citations we might rather exploit a reorganization of the data in the form of a tensor.

In this paper, we extend the use of the *Sparse Poisson Latent Block Model* to deal with sparse tensor data using jointly all information arising from documents. The proposed model is parsimonious and tailored for this kind of data. To estimate the parameters, we derive a suitable tensor co-clustering algorithm. Empirical results on several real-world text datasets highlight the advantages of our proposal which improves the clustering results of documents.

CCS CONCEPTS

• **Unsupervised learning**; • **Co-clustering** → *Tensor data*; • **Text mining**;

KEYWORDS

Co-clustering; Tensor data; Text mining.

ACM Reference Format:
Rafika Boutalbi[1,2], Lazhar Labiod[1], and Mohamed Nadif[1]. 2019. Sparse Tensor Co-Clustering as a Tool for Document Categorization. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331360

1 INTRODUCTION

Co-clustering which is a simultaneous clustering of both dimensions of a data matrix has proven to be more useful than traditional one-sided clustering especially when dealing with sparse data as is the case with document-term matrices. Generally, in document clustering, we rely on such matrices where each cell represents the occurrence of a word on a document. However, there is some additional available information like Keywords, co-authors, citations which not taken into account and it can improve the clustering results; two documents that have one or more authors in common and/or that quote each other, are likely to deal with the same topic. Incorporating this additional information leads us to consider a tensor representation of the data.

Despite the great interest for co-clustering and the tensor representation, few works tackle the co-clustering from tensor data. In fact, a large part of works are devoted mainly to popular factorization approaches such as `Tucker-decomposition` [15] and `PARAFAC` [8]. We can nevertheless mention the works related to our proposal such as the work of [2] based on Minimum Bregman information (`MBI`) to find co-clustering of a tensor. Most recently, in [16] the General Tensor Spectral Co-clustering (`GTSC`) method for co-clustering the modes of non-negative tensor has been developed. In [5] the authors proposed a tensor biclustering algorithm able to compute a subset of tensor rows and columns whose corresponding trajectories form a low-dimensional subspace. However, the majority of authors consider the same entities, for both sets of rows and columns, or do not consider the tensor co-clustering under a probabilistic approach.

To the best of our knowledge, this is the first attempt to formulate our objective when both sets -rows and columns- are different and with model-based co-clustering. To this end, we rely on the latent block model [7] for its flexibility to consider any data matrices. We propose a co-clustering model for sparse tensor data which can be viewed as a multi-way clustering model where each slice of the third dimension of the tensor represents a relation between two sets (see Figure 1). The goal is to simultaneously discover the row and column clusters and the relationship between these clusters for all slices.

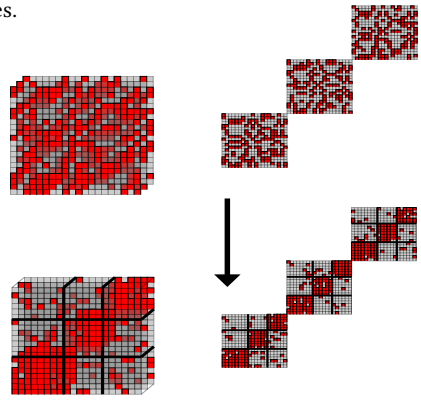

Figure 1: Goal of co-clustering for Sparse Tensor.

2 LATENT BLOCK MODEL (LBM)

Given an $n \times d$ data matrix $\mathbf{X} = (x_{ij}, i \in I = \{1, \ldots, n\}; j \in J = \{1, \ldots, d\})$. It is assumed that there exists a partition on I and a partition on J. A partition of $I \times J$ into $g \times m$ blocks will be represented by a pair of partitions $(\mathbf{Z}, \mathbf{W})$. The k-th row cluster corresponds to the set of rows i such that $z_{ik} = 1$ and $z_{ik'} = 0$ $\forall k' \neq k$. Thereby, the

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331360

partition $\mathbf{Z}$ can be represented by a matrix of elements in $\{0,1\}^g$ satisfying $\sum_{k=1}^{g} z_{ik} = 1$. Similarly, the ℓ-th column cluster corresponds to the set of columns j and the partition $\mathbf{W}$ can be represented by a matrix of elements in $\{0,1\}^m$ satisfying $\sum_{\ell=1}^{m} w_{j\ell} = 1$. Considering the Latent Block Model (LBM) [7], it is assumed that each element x_{ij} of the $k\ell$th block is generated according to a parameterized probability density function (pdf) $f(x_{ij};\alpha_{k\ell})$. Furthermore, in the LBM the univariate random variables x_{ij} are assumed to be conditionally independent given $(\mathbf{Z},\mathbf{W})$. Thereby, the conditional pdf of $\mathbf{X}$ can be expressed as $\prod_{i,j,k,\ell}\{f(x_{ij};\alpha_{k\ell})\}^{z_{ik}w_{j\ell}}$. From this hypothesis, we then consider the latent block model where the two sets I and J are considered as random samples and the row, and column labels become latent variables. Therefore, the parameter of the latent block model is $\boldsymbol{\Theta} = (\boldsymbol{\pi},\boldsymbol{\rho},\boldsymbol{\alpha})$, with $\boldsymbol{\pi} = (\pi_1,\ldots,\pi_g)$ and $\boldsymbol{\rho} = (\rho_1,\ldots,\rho_m)$ where $(\pi_k = P(z_{ik}=1), k = 1,\ldots,g)$, $(\rho_\ell = P(w_{j\ell}=1), \ell = 1,\ldots,m)$ are the mixing proportions and $\boldsymbol{\alpha} = (\alpha_{k\ell}; k = 1,\ldots g, \ell = 1,\ldots,m)$ where $\alpha_{k\ell}$ is the parameter of the distribution of block $k\ell$. Denoting $\mathcal{Z}$ and $\mathcal{W}$ the sets of possible labels $\mathbf{Z}$ for I and $\mathbf{W}$ for J, the pdf $f(\mathbf{X};\boldsymbol{\Theta})$ of $\mathbf{X}$ can be written

$$\sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,\ell} \rho_\ell^{w_{j\ell}} \times \prod_{i,j,k,\ell} \{f(x_{ij};\alpha_{k\ell})\}^{z_{ik}w_{j\ell}}. \quad (1)$$

Assuming that the complete data are the vector $(\mathbf{X},\mathbf{Z},\mathbf{W})$, i.e, we assume that the latent variables $\mathbf{Z}$ and $\mathbf{W}$ are known, the resulting complete data log-likelihood of the latent block model $L_C(\mathbf{Z},\mathbf{W},\boldsymbol{\Theta}) = \log f(\mathbf{X},\mathbf{Z},\mathbf{W};\boldsymbol{\Theta})$ can be written as follows

$$\sum_{i,k} z_{ik}\log\pi_k + \sum_{j,\ell} w_{j\ell}\log\rho_\ell + \sum_{i,j,k,\ell} z_{ik}w_{j\ell}\log f(x_{ij};\alpha_{k\ell}).$$

3 TENSOR LATENT BLOCK MODEL (TLBM)

Hereafter, we propose a novel Latent Block model for tensor data (TLBM). Few studies have addressed the issue of co-clustering for tensor data [5, 16]. Unlike classical LBM which considers data matrix $\mathbf{X} = [x_{ij}] \in \mathbb{R}^{n\times d}$, TLBM considers 3D data matrix $\mathbf{X} = [\mathbf{x}_{ij}] \in \mathbb{R}^{n\times d\times v}$ where n is the number of rows, d the number of columns, and v the number of covariates (slices). Figure 2 presents a tensor data with v slices. In this case, we can consider an extension of

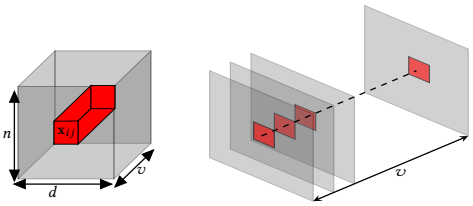

Figure 2: Data structure; $\mathbf{x}_{ij} = (x_{ij}^1,\ldots,x_{ij}^b,\ldots,x_{ij}^v)$

the latent block model (TLBM). The $\Phi(\mathbf{x}_{ij};\boldsymbol{\lambda}_{k\ell})$ is the pdf function, where $\boldsymbol{\lambda}_{k\ell}$ a $v\times 1$ vector formed by $(\lambda_{k\ell}^1,\ldots,\lambda_{k\ell}^b,\ldots,\lambda_{k\ell}^v)$ of the parameters of Φ. The complete data log-likelihood $L_C(\mathbf{Z},\mathbf{W},\boldsymbol{\Omega})$ is given by

$$\sum_{i,k} z_{ik}\log\pi_k + \sum_{j,\ell} w_{j\ell}\log\rho_\ell + \sum_{i,j,k,\ell} z_{ik}w_{j\ell}\log(\Phi(\mathbf{x}_{ij};\boldsymbol{\lambda}_{k\ell}))$$

where $\boldsymbol{\Omega}$ is formed by $\boldsymbol{\pi}$, $\boldsymbol{\rho}$ and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_{11},\ldots,\boldsymbol{\lambda}_{gm})$. Assuming the conditional independence per block which is here a cube, the third term of L_C becomes $\sum_{i,j,k,\ell} z_{ik}w_{j\ell}\left(\sum_{b=1}^{v}\log\Phi(x_{ij}^b;\lambda_{k\ell}^b)\right)$. To estimate $\boldsymbol{\Omega}$, the EM algorithm [3] is a candidate for this task. It maximizes the log-likelihood $f(\mathbf{X},\boldsymbol{\Omega})$ w.r. to $\boldsymbol{\Omega}$ iteratively by maximizing the conditional expectation of the complete data log-likelihood $L_C(\mathbf{Z},\mathbf{W};\boldsymbol{\Omega})$ w.r. to $\boldsymbol{\Omega}$, given a previous current estimate $\boldsymbol{\Omega}^{(c)}$ and the observed data $\mathbf{X}$. To solve this problem an approximation using the interpretation of the EM algorithm has been proposed; see, e.g., [6]. More precisely, the authors rely on the variational approach which consists in approximating the true likelihood by another expression using the following independence assumption: $P(z_{ik}=1,w_{j\ell}=1|\mathbf{X}) = P(z_{ik}=1|\mathbf{X})P(w_{j\ell}=1|\mathbf{X}) = \tilde{z}_{ik}\tilde{w}_{j\ell}$. Hence, the aim is to maximize the following lower bound of the log-likelihood criterion:

$$F_C(\tilde{\mathbf{Z}},\tilde{\mathbf{W}};\boldsymbol{\Omega}) = L_C(\tilde{\mathbf{Z}},\tilde{\mathbf{W}},\boldsymbol{\Omega}) + H(\tilde{\mathbf{Z}}) + H(\tilde{\mathbf{W}}) \quad (2)$$

where $H(\tilde{\mathbf{Z}}) = -\sum_{i,k}\tilde{z}_{ik}\log\tilde{z}_{ik}$, $H(\tilde{\mathbf{W}}) = -\sum_{j,\ell}\tilde{w}_{j\ell}\log\tilde{w}_{j\ell}$ and $L_C(\tilde{\mathbf{Z}},\tilde{\mathbf{W}},\boldsymbol{\Omega})$ is the fuzzy complete-data log-likelihood is given by

$$\sum_{i,k}\tilde{z}_{ik}\log\pi_k + \sum_{j,\ell}\tilde{w}_{j\ell}\log\rho_\ell + \sum_{i,j,k,\ell}\tilde{z}_{ik}\tilde{w}_{j\ell}\left(\sum_{b=1}^{v}\log\Phi(x_{ij}^b;\lambda_{k\ell}^b)\right). \quad (3)$$

4 TENSOR SPARSE POISSON LBM (TSPLBM)

Recently, in [1, 11] the authors proposed a generative mixture model for co-clustering document-term matrices. With SPLBM, the authors first assume that for each diagonal block kk the values $x_{ij}\sim\mathcal{P}(\lambda_{ij})$ where $\lambda_{ij} = x_{i.}x_{.j}\sum_k[z_{ik}w_{jk}]\gamma_{kk}$ with $x_{i.} = \sum_j x_{ij}$ and $x_{.j} = \sum_i x_{ij}$. Second, they assume that for each block $k\ell$ with $k\neq\ell$, $x_{ij}\sim\mathcal{P}(\lambda_{ij})$ where the parameter λ_{ij} takes the following form : $\lambda_{ij} = x_{i.}x_{.j}\sum_{k,\ell\neq k}[z_{ik}w_{j\ell}]\gamma$. Assuming $\forall\ell\neq k,\gamma_{k\ell}=\gamma$ leads to suppose that all blocks outside the diagonal share the same parameter. SPLBM has been designed from the ground up to deal with data sparsity problems. As a consequence, in addition to seeking homogeneous blocks, it also filters out homogeneous but noisy ones due to the sparsity of the data.

In the following we propose to extend SPLBM to deal with tensor data. It is easy to show that $\sum_{i,j,k}\tilde{z}_{ik}\tilde{w}_{jk}\left(\sum_{b=1}^{v}\log\Phi(x_{ij}^b;\lambda_{k\ell}^b)\right)$ (the third term of $L_C(\tilde{\mathbf{Z}},\tilde{\mathbf{W}},\boldsymbol{\Omega})$ (3)) takes the following form

$$\sum_{i,j,k}\tilde{z}_{ik}\tilde{w}_{jk}\left(\sum_{b=1}^{v}\log\Phi(x_{ij}^b;\lambda_{kk}^b)\right) + \sum_{i,j,k,\ell\neq k}\tilde{z}_{ik}\tilde{w}_{j\ell}\left(\sum_{b=1}^{v}\log\Phi(x_{ij}^b;\lambda^b)\right).$$

For each block $k=1,\ldots,g$ and each slice b, the x_{ij}^b's are distributed according $\mathcal{P}(x_{i.}^b x_{.j}^b\gamma_{kk}^b)$ and outside according $\mathcal{P}(x_{i.}^b x_{.j}^b\gamma^b)$. After some algebraic calculations and simplifications, (3) becomes (up a constant)

$$\sum_{i,k}\tilde{z}_{ik}\log\pi_k + \sum_{j,k}\tilde{w}_{jk}\log\rho_k + \sum_b\sum_k(x_{kk}^b\log(\gamma_{kk}^b) - x_{k.}^b x_{.k}^b\gamma_{kk}^b)$$
$$+\sum_b\left((N_b - \sum_k x_{kk}^b)\log(\gamma^b) - (N_b^2 - \sum_k x_{k.}^b x_{.k}^b)\gamma^b\right)$$
$$= \sum_{i,k}\tilde{z}_{ik}\log\pi_k + \sum_{j,k}\tilde{w}_{jk}\log\rho_k$$
$$+\sum_b\left(\sum_k\left[x_{kk}^b\log(\frac{\gamma_{kk}^b}{\gamma^b}) - x_{k.}^b x_{.k}^b(\gamma_{kk}^b - \gamma^b)\right] + N_b(\log(\gamma) - N_b\gamma)\right),$$

where $x_{k.}^b = \sum_i\tilde{z}_{ik}x_{i.}^b$, $x_{.k}^b = \sum_j\tilde{w}_{jk}x_{.j}^b$, $x_{kk}^b = \sum_{i,j}\tilde{z}_{ik}\tilde{w}_{jk}x_{ij}^b$ and $N_b = \sum_{i,j}x_{ij}^b$.

5 VARIATIONAL EM ALGORITHM

In what follows, we detail the Expectation (E) and Maximization (M) step of the Variational EM algorithm for tensor data. The E-step consists in computing, for all i, j, k the posterior probabilities $\tilde{z}_{ik}$ and $\tilde{w}_{jk}$ maximizing F_C given the estimated parameters Ω. As $\sum_k \tilde{z}_{ik} = 1$ and $\sum_k \tilde{w}_{jk} = 1$, using the corresponding Lagrangians, up to terms which are not function of $\tilde{z}_{ik}$ and $\tilde{w}_{jk}$ leads to

$$\tilde{z}_{ik} \propto \pi_k \exp\left(\sum_j \tilde{w}_{jk} \sum_{b=1}^{v} x_{ij}^b \log(\frac{\gamma_{kk}^b}{\gamma^b})\right),$$

$$\tilde{w}_{jk} \propto \rho_k \exp\left(\sum_i \tilde{z}_{ik} \sum_{b=1}^{v} x_{ij}^b \log(\frac{\gamma_{kk}^b}{\gamma^b})\right).$$

Given the previously computed posterior probabilities $\tilde{\mathbf{Z}}$ and $\tilde{\mathbf{W}}$, the M-step consists in updating , $\forall k$, the parameters π_k, ρ_k, $\boldsymbol{\gamma}_{kk}^b$ and $\boldsymbol{\gamma}^b$ maximizing $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \Omega)$. The estimated parameters are defined as follows. First, taking into account the constraints $\sum_k \pi_k = 1$ and $\sum_k \rho_k = 1$, it is easy to show that $\pi_k = \frac{\sum_i \tilde{z}_{ik}}{n}$ and $\rho_k = \frac{\sum_j \tilde{w}_{jk}}{d}$. Secondly, it is easy to derive

$$\boldsymbol{\gamma}_{kk}^b = \frac{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{jk}x_{ij}^b}{\sum_i \tilde{z}_{ik}x_{i.}^b \sum_j \tilde{w}_{jk}x_{.j}^b} = \frac{x_{kk}^b}{x_{k.}^b x_{.k}^b} \text{ and,}$$

$$\boldsymbol{\gamma}^b = \frac{N_b - \sum_{i,j,k} \tilde{z}_{ik}\tilde{w}_{jk}x_{ij}^b}{N_b^2 - \sum_k \sum_i \tilde{z}_{ik}x_{i.}^b \sum_j \tilde{w}_{jk}x_{.j}^b} = \frac{N_b - \sum_k x_{kk}^b}{N_b^2 - \sum_k x_{k.}^b x_{.k}^b}.$$

The proposed algorithm for sparse tensor (ST) data, referred to as VEM-ST in Algorithm 1, alternates the two previously described steps Expectation-Maximization. At the convergence, a hard co-clustering is deduced from $\tilde{z}_{ik}$'s and $\tilde{w}_{jk}$'s using the maximum a posteriori principle.

Algorithm 1: VEM-ST

Input: $\mathbf{X}$, g.
Initialization $(\mathbf{Z}, \mathbf{W})$ randomly, compute Ω
repeat
  E-Step : Compute $\tilde{z}_{ik}$ and $\tilde{w}_{jk}$
  - $\tilde{z}_{ik} \propto \pi_k \exp\left(\sum_j \tilde{w}_{jk} \sum_{b=1}^{v} x_{ij}^b \log(\frac{\gamma_{kk}^b}{\gamma^b})\right)$
  - $\tilde{w}_{jk} \propto \rho_k \exp\left(\sum_i \tilde{z}_{ik} \sum_{b=1}^{v} x_{ij}^b \log(\frac{\gamma_{kk}^b}{\gamma^b})\right)$

  M-Step : Update Ω
until *convergence*;
return Z, W

6 EXPERIMENTS

In our experiments, we aim to evaluate VEM-ST in terms of document clustering leading to measure the impact of additional information. Thereby, we compare VEM-ST with Spherical K-means, Itcc [4], and VEM-ST$_b$ applied on each slice b of tensor and three other algorithms applied on tensor data namely Tucker-decomposition [9], PARAFAC [9] and GTSC [16]. Note that the first two are used with ranks number equals to 10 and followed by K-means. We perform 30 random initialisations, and compute the Accuracy (ACC), Adjusted Rand index (ARI) [13] and Normalized Mutual Information (NMI) [14] metrics.

We use three text datasets DBLP1, DBLP2 and PubMed Diabetes[1] to highlight the objective of the proposed algorithm. DBLP1 and DBLP2 are constructed from DBLP[2], by selecting three journals for each one. The selected journals for DBLP1 are SIGMOD, STOC, and SIGIR. The journals selected for DBLP2 are Discrete Applied Mathematics, IEEE software, and SIGIR. For PubMed Diabetes dataset the papers are categorized into three types, the first one deals with Diabetes mellitus of type 1, the second with Diabetes mellitus of type 2, and the third with Diabetes mellitus Experimental. We extract from these different datasets information linked documents:

- Co-terms matrix on the title: each cell represents the number of times that a term is present simultaneously in the title of a pair of papers,
- Co-terms matrix on the abstract: each cell represents the number of times that a pair of papers share a term extracting from abstract,
- Co-authors matrix: each cell represents the number of common authors of a pair of papers,
- Citations matrix: is a binary data matrix where 1 indicates the presence of a citation between two papers.

The constructed tensor ($Paper \times Paper \times Relation$) for each dataset DBLP1, DBLP2 and PubMed Diabetes has respectively size ($2223 \times 2223 \times 4$), ($1949 \times 1949 \times 4$), and ($4354 \times 4354 \times 4$) and different rates of sparsity 0.93, 0.94, and 0.69 respectively. In Figures 3, 4 and 5, the first plots represent the low-dimensional projection of papers from tensor data of each dataset using the *Multiple Factor Analysis* (MFA). MFA deals with a multiple table where the slices are contingency tables [10]. We notice that the three datasets have different degree of complexity. On the other hand, the four other plots of each figure represent the slices of tensor namely Co-terms matrices on the titles and abstracts, Co-authors and Citations matrices. In Table 1 are reported the performances of the seven algorithms (cited above) on the three datasets. In terms of ACC, NMI and ARI, we observe in most cases, that VEM-ST is better than other algorithms applied on each slice and those applied on tensor data. We observe that PubMed Diabets which is the least sparse dataset, we obtain the lowest results for the three measures ACC, NMI and ARI due to the complex structure of dataset appearing on figure 5. Further note that GTSC, less effective than VEM-ST, reaches better results than PARAFAC and Tucker-decomposition followed by K-means.

7 CONCLUSION

To deal with document categorization with additional information like key-words, co-authors and citations, we proposed a data structure including all information and a novel model-based co-clustering on tensor data. The proposed model is an extension of the latent block model (LBM). We have derived a suitable sparse tensor Poisson co-clustering model (STLBM). To estimate the parameters, a variational EM algorithm is developed, referred to as VEM-ST. VEM-ST does a better job than GTSC, and other competitive algorithms applied on each slice of tensor data.

Due to the flexibility of the proposed model, our findings open up interesting opportunities for other applications like relational

[1] https://linqs.soe.ucsc.edu/data
[2] https://aminer.org/citation

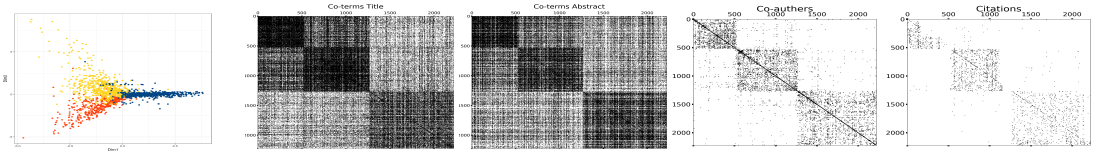

Figure 3: DBLP 1

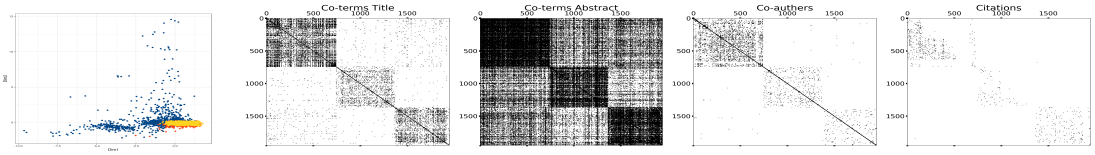

Figure 4: DBLP 2

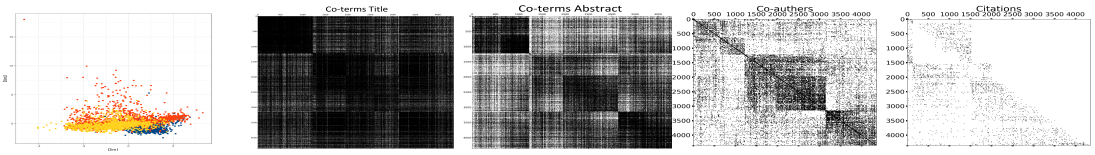

Figure 5: PubMed Diabets

Table 1: Evaluation of documents clustering in terms of ACC, NMI and ARI for the three datasets

Datasets	Metrics	ACC				NMI				ARI			
	Algorithms	Titles	Abstracts	Co-authors	Citations	Titles	Abstracts	Co-authors	Citations	Titles	Abstracts	Co-authors	Citations
DBLP1	Spherical K-means	0.75 ± 0.0	0.79 ± 0.0	0.4 ± 0.0	0.43 ± 0.0	0.38 ± 0.0	0.41 ± 0.0	0.02 ± 0.0	0.03 ± 0.0	0.45 ± 0.0	0.48 ± 0.0	0.02 ± 0.0	0.02 ± 0.0
	Itcc	0.72 ± 0.0	0.79 ± 0.0	0.41 ± 0.0	0.41 ± 0.0	0.36 ± 0.0	0.41 ± 0.0	0.02 ± 0.0	0.02 ± 0.0	0.42 ± 0.0	0.49 ± 0.0	0.02 ± 0.0	0.01 ± 0.0
	VEM-ST$_b$	0.72± 0.0	0.72 ± 0.0	0.43 ± 0.0	0.42 ± 0.0	0.38 ± 0.0	0.39 ± 0.0	0.02 ± 0.0	0.02 ± 0.0	0.43 ± 0.0	0.43 ± 0.0	0.02 ± 0.0	0.01 ± 0.0
	PARAFAC[1]	0.54 ± 0.01				0.10 ± 0.0				0.08 ± 0.0			
	Tucker-decomp[1]	0.55 ± 0.0				0.11 ± 0.0				0.04 ± 0.0			
	GTSC[2]	0.87 ± 0.0				**0.61 ± 0.0**				0.65 ± 0.0			
	VEM-ST	**0.89 ± 0.0**				**0.61 ± 0.0**				**0.70 ± 0.0**			
DBLP2	Spherical K-means	0.52 ± 0.0	0.69 ± 0.0	0.37 ± 0.0	0.41 ± 0.0	0.13 ± 0.0	0.30 ± 0.0	0.01 ± 0.0	0.05 ± 0.0	0.09 ± 0.0	0.28 ± 0.0	0.01 ± 0.0	0.01 ± 0.0
	Itcc	0.43 ± 0.0	0.80 ± 0.0	0.37 ± 0.0	0.39 ± 0.0	0.05 ± 0.0	0.45 ± 0.0	0.03 ± 0.0	0.04 ± 0.0	0.04 ± 0.0	0.5 ± 0.0	0.01 ± 0.0	0.01 ± 0.0
	VEM-ST$_b$	0.45 ± 0.0	0.80 ± 0.0	0.38 ± 0.0	0.39 ± 0.0	0.05 ± 0.0	0.47 ± 0.0	0.01 ± 0.0	0.02 ± 0.0	0.05 ± 0.0	0.5 ± 0.0	0.01 ± 0.0	0.0 ± 0.0
	PARAFAC[1]	0.65 ± 0.0				0.16 ± 0.0				0.11 ± 0.0			
	Tucker-decomp[1]	0.62 ± 0.0				0.14 ± 0.0				0.07 ± 0.0			
	GTSC[2]	0.55 ± 0.0				0.27 ± 0.0				0.25 ± 0.0			
	VEM-ST	**0.81 ± 0.0**				**0.55 ± 0.0**				**0.56 ± 0.0**			
PubMed Diabets	Spherical K-means	0.54 ± 0.0	**0.64 ± 0.0**	0.38 ± 0.0	0.43 ± 0.0	0.18 ± 0.0	0.29 ± 0.0	0.01 ± 0.0	0.0 ± 0.0	0.17 ± 0.0	0.26 ± 0.0	0.01 ± 0.0	0.0 ± 0.0
	Itcc	0.55 ± 0.0	**0.64 ± 0.0**	0.37 ± 0.0	0.43 ± 0.0	0.18 ± 0.0	0.3 ± 0.0	0.01 ± 0.0	0.0 ± 0.0	0.16 ± 0.0	0.28 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	VEM-ST$_b$	0.54 ± 0.0	0.61 ± 0.0	0.37 ± 0.0	0.44 ± 0.0	0.19 ± 0.0	0.3 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.18 ± 0.0	0.28 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	PARAFAC[1]	0.54 ± 0.0				0.14 ± 0.0				0.10 ± 0.0			
	Tucker-decomp[1]	0.53 ± 0.0				0.10 ± 0.0				0.05 ± 0.0			
	GTSC[2]	0.53 ± 0.0				0.26 ± 0.0				0.24 ± 0.0			
	VEM-ST	**0.64 ± 0.0**				**0.33 ± 0.0**				**0.31 ± 0.0**			

[1] PARAFAC and Tucker-decomp followed by K-means,
[2] GTSC with appropriate parameters to obtain the desired number of co-clusters.

learning. Furthermore, other block models relying on other appropriate distributions can be thought for tensor data; see, e.g., [12].

REFERENCES

[1] Ailem, M., Role, F., and Nadif, M. Sparse poisson latent block model for document clustering. *IEEE Transactions on Knowledge and Data Engineering 29*, 7 (2017), 1563–1576.

[2] Banerjee, A., Krumpelman, C., Ghosh, J., Basu, S., and Mooney, R. J. Model-based overlapping clustering. In *Proceedings of the Eleventh ACM SIGKDD* (2005), pp. 532–537.

[3] Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B 39* (1977), 1–38.

[4] Dhillon, I. S., Mallela, S., and Modha, D. S. Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD* (2003), pp. 89–98.

[5] Feizi, S., Javadi, H., and Tse, D. Tensor biclustering. In *Advances in Neural Information Processing Systems 30* (2017), Curran Associates, Inc., pp. 1311–1320.

[6] Govaert, G., and Nadif, M. An EM algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and machine intelligence 27*, 4 (2005), 643–647.

[7] Govaert, G., and Nadif, M. *Co-clustering*. Wiley-IEEE Press, 2013.

[8] Harshman, R. A., and Lundy, M. E. Parafac : parallel factor analysis. *Computational statistics and data analysis 18* (1994), 39–72.

[9] Kossaifi, J., Panagakis, Y., Anandkumar, A., and Pantic, M. Tensorly: Tensor learning in python. *CoRR abs/1610.09555* (2018).

[10] Pagès, J. *Multiple factor analysis by example using R*. Chapman and Hall/CRC, 2014.

[11] Role, F., Morbieu, S., and Nadif, M. Coclust: A python package for co-clustering. *Journal of Statistical Software 88*, 7 (2019), 1–29.

[12] Salah, A., and Nadif, M. Directional co-clustering. *Advances in Data Analysis and Classification* (2018), 1–30.

[13] Steinley, D. Properties of the hubert-arabie adjusted rand index. *Psychological methods 9*, 3 (2004), 386.

[14] Strehl, A., and Ghosh, J. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research 3* (2002), 583–617.

[15] Tucker, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika 31*, 3 (1966), 279–311.

[16] Wu, T., Benson, A. R., and Gleich, D. F. General tensor spectral co-clustering for higher-order data. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016, pp. 2559–2567.