Model-based tensor (co)-clustering and applications

Rafika Boutalbi

DOCTORAL SCHOOL OF COMPUTER SCIENCE, TELECOMMUNICATION AND ELECTRONICS (EDITE) PARIS

MODEL-BASED TENSOR (CO)-CLUSTERING AND APPLICATIONS

by

Rafika Boutalbi

THIS DISSERTATION IS SUBMITTED FOR THE DEGREE OF
DOCTOR OF COMPUTER SCIENCE
AT UNIVERSITÉ DE PARIS

COMMITTEE:

Pr. Mohamed Nadif	Université de Paris, Supervisor
Dr. Lazhar Labiod	Université de Paris, Co-supervisor
Pr. Allou Samé	Université Paris-Est
Pr. Pierre-François Marteau	Université de Bretagne Sud
Pr. Rafik Abdesselam	Université Lyon 2
Pr. Nadia Ghazzali	Université Trois rivières
Mr Dan Dassier	Trinov

ÉCOLE DOCTORALE INFORMATIQUE, TÉLÉCOMMUNICATIONS ET ÉLECTRONIQUE
(EDITE) DE PARIS

CLASSIFICATION CROISÉE DE DONNÉES TENSORIELLES ET APPLICATIONS

par

Rafika Boutalbi

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION DU TITRE DE
DOCTEUR EN INFORMATIQUE
À UNIVERSITÉ DE PARIS

JURY COMPOSÉ DE :

DIRECTEUR DE THÈSE	:	Mr. Mohamed Nadif	(Professeur, Université de Paris)
CO-ENCADRANT	:	Mr Lazhar Labiod	(MCF, Université de Paris)
RAPPORTEUR	:	Mr Allou Same	(Research, HDR, Université Paris-Est)
RAPPORTEUR	:	Mr Pierre-François Marteau	(Professeur, Université Bretagne Sud)
EXAMINATEUR	:	Mr Rafik Abdesselam	(Professeur, Université Lyon 2)
EXAMINATEUR	:	Mme Nadia Ghazzali	(Professeur, Université Trois rivières, Canada)
INVITÉ	:	Mr Dan Dassier	(Trinov)

"*Le modèle doit suivre les données et non l'inverse.*"

Jean-Paul Benzécri, 1973

Résumé

La classification non supervisée ou *clustering* suscite un grand intérêt dans la communauté d'apprentissage machine. Etant donné un ensemble d'objets décrits par un ensemble d'attributs, le *clustering* vise à partitionner l'ensemble des objets en classes homogènes. Le regroupement ou catégorisation de cet ensemble, est souvent nécessaire pour le traitement de données massives, devenu actuellement un axe de recherche prioritaire. A noter que lorsqu'on s'intéresse au clustering, nous faisons généralement référence au clustering de l'ensemble des objets. Depuis deux décennies, un intérêt est porté à la classification croisée (ou *co-clustering*) qui permet de regrouper simultanément les lignes et les colonnes d'une matrice de données. Le *co-clustering* conduit de ce fait à une réorganisation des données en blocs homogènes (après permutations appropriées). Cette approche joue un rôle important dans une grande variété d'applications où les données sont généralement organisées dans des tableaux à double entrées [Govaert and Nadif, 2013]. Cependant si on considère l'exemple du *clustering* d'articles, nous pouvons collecter plusieurs informations telles que les termes en commun, les co-auteurs et les citations, qui conduisent naturellement à une représentation tensorielle. L'exploitation d'un tel tenseur d'ordre 3 permettrait d'améliorer les résultats de clustering d'un des ensembles. Ainsi, deux articles qui partagent un ensemble important de mots en commun, qui ont des auteurs en commun et qui partagent une bibliographie commune, sont très susceptibles de traiter d'une même thématique. Dans cette thèse nous nous intéressons à de telles structures de données. Malgré le grand intérêt pour le *co-clustering* et la représentation tensorielle, peu de travaux portent sur le *co-clustering* de tenseurs. Nous pouvons néanmoins citer le travail basé sur l'information Minimum Bregman (MBI) [Banerjee et al., 2005], ou encore la méthode de *co-clustering* de tenseurs non négatifs GTSC (General Tensor Spectral Co-Clustering) [Wu et al., 2016]. Mais la majorité des travaux considèrent le co-clustering à partir de méthodes de factorisation tensorielles. Dans cette thèse nous proposons de nouvelles approches probabilistes pour le co-clustering de tenseur d'ordre 3.

Dès lors plusieurs défis sont à relever dont les suivants. Comment gérer efficacement les données de grande dimension? Comment gérer la *sparsité* des données et exploiter les dépendances inter-tranches des données tensorielles? S'inspirant de la célèbre citation de Jean Paul Benzcri *"Le modèle doit suivre les données et non l'inverse"*, nous avons choisi dans cette thèse de nous appuyer sur des modèles de mélange appropriés. Ainsi, nos contributions sont basées sur le modèle des blocs latents ou *(LBM, Latent Block Model)* pour le *co-clustering*, proposé pour la première fois par Govaert and Nadif [2003].

Voici une brève description des différentes contributions: a) Extension du formalisme des *LBM* au *co-clustering* des données tensorielles et présentation d'un nouveau modèle *Tensor LBM(TLBM)* comme solution, b) Proposition d'un *Sparse TLBM* prenant en compte la *sparsité* et son extension pour la gestion des graphes multiples ou graphes multi-vues, et c) Développement d'une méthode de *co-clusterwise* qui intègre le *co-clustering* dans un cadre d'apprentissage supervisé. Ces contributions ont été évaluées avec succès sur des données tensorielles issues de divers domaines allant des systèmes de recommandation, le *clustering* d'images hyperspectrales, la catégorisation de documents, à l'optimisation de la gestion des déchets. Elles permettent également d'envisager des pistes de recherches futures intéressantes et immédiates. Par exemple, l'extension du modèle proposé au *tri-clustering* et aux séries temporelles multivariées.

Abstract

Clustering, which seeks to group together similar data points according to a given criterion, is an important unsupervised learning technique to deal with large scale data. In particular, given a data matrix where rows represent objects and columns represent features, clustering aims to partition only one dimension of the matrix at a time, by clustering either objects or features. Although successfully applied in several application domains, clustering techniques are often challenged by certain characteristics exhibited by some datasets such as high dimensionality and sparsity. When it comes to such data, co-clustering techniques, which allow the simultaneous clustering of rows and columns of a data matrix, has proven to be more beneficial. In particular, co-clustering techniques allow the exploitation of the inherent duality between the objects set and features set, which make them more effective even if we are interested in the clustering of only one dimension of our data matrix. In addition, co-clustering turns out to be more efficient since compressed matrices are used at each time step of the process instead of the whole matrix for traditional clustering.

Although co-clustering approaches have been successfully applied in a variety of applications, existing approaches are specially tailored for datasets represented by double-entry tables. However, in several real-world applications, two dimensions are not sufficient to represent the dataset. For example, if we consider the articles clustering problem, several information linked to the articles can be collected, such as common words, co-authors and citations, which naturally lead to a tensorial representation. Intuitively, leveraging all these information would lead to a better clustering quality. In particular, two articles that share a large set of words, authors and citations are very likely to be similar. Despite the great interest of tensor co-clustering models, research works are extremely limited in this context and rely, for most of them, on tensor factorization methods.

Inspired by the famous statement made by Jean Paul Benzécri "The model must follow the data and not vice versa", we have chosen in this thesis to rely on appropriate mixture models. More explicitly, we propose several new co-clustering models which are specially tailored for tensorial representations as well as robust towards data sparsity. Our contribution can be summarized as follows. First, we propose to extend the LBM (Latent Block Model) formalism to take into account tensorial structures. More specifically, we present Tensor LBM (TLBM), a powerful tensor co-clustering model that we successfully applied on diverse kind of data. Moreover, we highlight that the derived algorithm `VEM-T`, reveals the most meaningful co-clusters from tensor data. Second, we develop a novel Sparse TLBM taking into account sparsity. We extend its use for the management of multiple graphs (or multi-view graphs), leading to implicit consensus clustering of multiple graphs. As a last contribution of this thesis, we propose a new co-clusterwise method which integrates co-clustering in a supervised learning framework. These contributions have been successfully evaluated on tensorial data from various fields ranging from recommendation systems, clustering of hyperspectral images and categorization of documents, to waste management optimization. They also allow us to envisage interesting and immediate future research avenues. For instance, the extension of the proposed models to tri-clustering and multivariate time series.

Acknowledgements

La réalisation de cette thèse a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude. Je voudrais tout d'abord exprimer ma reconnaissance à mon directeur de thèse Mr Mohamed Nadif, professeur à l'université Paris Descartes et directeur de l'équipe MLDS du LIPADE pour sa patience, sa disponibilité, son dévouement et ses judicieux conseils qui ont contribué à enrichir ma réflexion. Je remercie aussi mon co-encadrant Mr Lazhar Labiod, pour son apport et son aide précieuse notamment lors de la rédaction de ce mémoire. Je les remercie d'avoir partagé avec moi leurs connaissances et leurs expériences, tout en m'accordant une confiance et une large indépendance dans l'élaboration de cette thèse.

Je tiens à remercier spécialement Mr Dan Dassier président de l'entreprise Trinov pour son soutien tout au long de ma démarche. Mais aussi ses collaborateurs, mes collègues Khaled, Thibault Lepelletier et Balsam pour leur accueil et les conditions de travail privilégiées qu'ils m'ont offert. J'adresse mes sincères remerciement à l'ensemble des personnes inspirantes, ingénieuses et pleines de compétences qui m'ont entouré au sein de Trinov pendant ces trois années, et qui m'ont permis de mieux comprendre le fonctionnement passé et actuel de ce secteur. À Thibault Caporal, Abdallah, Gawain, Hakim, Raouf, Thibaut Des Closets, Dian, Pafi, Meriam, Thibault Defourneau, Cecille, Mahé à qui j'aurais tant aimé lire cette thèse.

Je remercie les rapporteurs de cette thèse, monsieur Allou Same et monsieur Pierre-François Marteau pour leur lecture, et l'intérêt qu'ils ont porté à mon travail. Je remercie également les autres membres du jury, monsieur Rafik Abdesselam ainsi que madame Nadia Ghazzali qui ont accepté de juger ce travail.

Je souhaite remercier spécialement mes deux amies Mélissa et Afifa qui m'ont aidé lors de la relecture de cette thèse. Mes remerciements vont aussi à mes collègues de bureau Aghiles, Mickael et Stanislas, qui ont partagé avec moi cette aventure et notre bureau. Merci pour toutes nos discussions et les repas pris ensemble.

Cette thèse je la dédie en premier lieu à mes parents, qui m'ont tant appris de la vie. Je les remercie de m'avoir soutenue dans toutes mes épreuves et pour leurs encouragements. Je dédie aussi cette thèse à mes deux sœurs Narjess et Karima qui sont non seulement mes sœurs, mais aussi mes amies et confidantes. Merci d'avoir été là tout le temps et de m'avoir soutenue. Vous êtes exceptionnelles.

J'adresse mes remerciements à mes tata, Akila, Horia, Adraa, Saida et à mes tontons Hocine et Chérif. Merci beaucoup pour votre soutien, pour votre bienveillance envers moi. Vous n'avez cessez de m'encourager et de me considérer comme votre fille. Je ne vous en serez jamais assez reconnaissante.

J'adresse mes remerciements à tous mes amis. Tout d'abord à Oussama, mon meilleur ami, on se connaît déjà depuis presque 10 ans, tu n'as jamais cessé d'être à mes cotés et me soutenir, je ne t'en serais jamais assez reconnaissante. À mes copines Amina, Sarra, Amel, Trang, Rim et Inés, je vous remercie pour votre présence et votre soutien.

Je souhaite enfin remercier toutes les personnes qui ont fait partie de ma vie pendant ces trois années de thèse. Chacun d'entre vous a contribué à maintenir ma motivation afin de donner le meilleure de moi même en m'impliquant dans cette thèse. Enfin cette aventure fut exceptionnelle, avec bien sur des moments compliqués, mais très enrichissant.

Contents

Abstract v

Acknowledgements ix

Introduction 1
Motivation . . . 2
Contribution . . . 3
Overview . . . 4
Publications . . . 5

1 (Co)-clustering of two-way and three-way tensor data 7
1.1 Clustering . . . 7
1.1.1 Hierarchical clustering . . . 7
Agglomerative methods . . . 8
Divisive methods . . . 8
1.1.2 Density-based approaches . . . 8
1.1.3 Graph-based approaches . . . 9
1.1.4 Partitional clustering approaches . . . 10
Centroid-based approaches . . . 10
Mixture-based approaches . . . 11
1.1.5 Clustering evaluation metrics . . . 12
1.2 Co-clustering . . . 13
1.2.1 Metric-based approaches . . . 13
1.2.2 Graph-based approaches . . . 14
1.2.3 Matrix factorization-based approaches . . . 14
1.2.4 Model-based approaches . . . 14
1.3 Clustering and analysis of Tensor data . . . 15
1.3.1 Notation and Preliminaries . . . 15
Tensor representation . . . 15
Tensor proprieties . . . 17
Tensor products . . . 17
Transforming tensor to matrix . . . 18
1.3.2 Tensor analysis and clustering approaches . . . 19
Tensor factorization based approaches . . . 19
Stochastic approaches. . . . 20
Low-rank approximation based approaches . . . 21
1.3.3 Tensor clustering applications . . . 21
Signal Processing . . . 21

Images and Hyperspectral images . . . 22
Recommender systems . . . 22
Other applications . . . 22
1.4 Conclusion . . . 23

2 Latent Block Model for Tensor Data **25**
2.1 Introduction . . . 25
2.2 Extension of Latent block model for tensors . . . 26
2.2.1 Latent block model . . . 26
2.2.2 Latent Block Model for Tensor data (TLBM) . . . 29
2.3 Variational EM algorithm for TLBM . . . 31
2.3.1 E-step . . . 32
2.3.2 M-step . . . 33
2.4 Classification Maximum Likelihood approach . . . 34
2.5 Experimental results . . . 36
2.5.1 Synthetic datasets and Competitive methods . . . 36
2.5.2 Real datasets . . . 38
Recommender system application . . . 38
Multi-spectral images analysis . . . 40
Document categorization . . . 41
2.6 Conclusion . . . 44

3 Sparse Poisson Tensor Co-clustering **45**
3.1 Introduction . . . 45
3.2 Sparse Tensor Co-Clustering . . . 46
3.2.1 Sparse Poisson LBM (SPLBM) . . . 46
3.2.2 Tensor Sparse Poisson LBM (TSPLBM) . . . 47
3.2.3 Variational EM algorithm . . . 49
3.3 Clustering from Multiple Graphs . . . 50
3.3.1 Related Work . . . 51
3.3.2 Poisson Latent and Stochastic Block Models . . . 52
3.3.3 TSPLBM with multiple graphs . . . 53
3.3.4 Variational Inference . . . 54
3.4 Experiments . . . 55
3.4.1 Datasets and evaluation . . . 55
3.4.2 Algorithm evaluation . . . 57
3.4.3 Implicit consensus VS explicit consensus . . . 59
3.5 Conclusion . . . 63

4 Latent Block Regression Model **65**
4.1 Introduction . . . 65
4.2 Recommendation systems . . . 67
4.3 From Clusterwise regression to co-clusterwise regression . . . 68
4.3.1 Co-clustering and LBM . . . 68
4.3.2 Latent Block Regression Model (LBRM) . . . 68
4.4 Variational EM algorithm . . . 70

4.5 Experimental results . . . 71
4.5.1 Simulation study . . . 72
4.5.2 Illustrative example . . . 73
4.5.3 Recommender system application . . . 76
4.6 Conclusion . . . 81

5 Using Tensor Analysis for Original Applications **83**
5.1 Waste management applications . . . 83
5.1.1 Recommendation system for waste management . . . 85
5.1.2 Waste collection trajectories . . . 87
Store clustering . . . 88
Pathfinding optimization . . . 89
5.1.3 Markdowns analysis . . . 92
5.2 Analysis of EGC conference evolution . . . 94
5.2.1 Data preprocessing and description . . . 94
5.2.2 Topic modeling of papers . . . 95
5.2.3 Analysis of authors' communities . . . 98
5.2.4 Recommendation of the *lecture committee* . . . 100
5.3 Conclusion . . . 102

Conclusion and Perspectives **103**

A Updating of Common Parameters of Tensor Models **105**
A.1 Update $\tilde{z}_{ik}$ and $\tilde{w}_{j\ell}$ $\forall i,k,j,\ell$ for TLBM . . . 105
A.2 Estimation of the π_k and ρ_ℓ $\forall k,\ell$ of TLBM . . . 106

B Estimation of TLBM's Parameters **107**
B.1 Estimation of the $\boldsymbol{\mu}_{k\ell}$ and $\Sigma_{k\ell}$ $\forall k,\ell$ parameters of Gaussian TLBM . . . 107
B.2 Estimation of the $\boldsymbol{\mu}^b_{k\ell}$'s of Bernoulli TLBM . . . 108
B.3 Estimation of the $\gamma^b_{k\ell}$'s of Poisson TLBM . . . 109

C Estimation of TSPLBM's Parameters **111**
C.1 Estimation of the γ^b_{kk} parameter . . . 111
C.2 Estimation of the γ^b parameter . . . 111

D Estimation of LBRM's Parameters **113**
D.1 Estimation of the $\boldsymbol{\beta}_{k\ell}$ parameter . . . 113
D.2 Estimation of the $\sigma^2_{k\ell}$ parameter . . . 114

Bibliography **115**

List of Figures

1 Examples of tensor data representation. 2

1.1 An hierarchical clustering represented by a dendrogram. 8
1.2 Political blogs communities. 9
1.3 Mixture of two Gaussian density functions. 11
1.4 (left) Original matrix, (middle) Rows clustering results, and (right) Co-clustering results. 13
1.5 Third-way tensor data representation. 16
1.6 Different representations of tensor elements. 16
1.7 Slices representations of tensor. 16
1.8 *Cube* tensor with ones on diagonal. 17
1.9 Tensor compression: (A) Three-way tensor, (B) Vertical compression, (C) Horizontal compression, (D) Frontal compression. 19

2.1 Goal of co-clustering for Binary Tensor data. 26
2.2 LBM graphical model. 27
2.3 (a) Data structure, (b) Gaussian LBM, (c) Gaussian TLBM. 29
2.4 Simulated binary datasets. 36
2.5 Simulated continuous datasets. 37
2.6 Distribution of the centers $\mu_{k\ell}$ for all co-clusters. 39
2.7 (a) Co-clustering data matrix, (b) Distribution of Age per row clusters. 39
2.8 (a) The four cells type, (b) Example of multispectral image from dataset. . . . 40
2.9 Co-clustering matrix of different slice of features. 41
2.10 Obtained results using Multiple Factor Analysis. 42
2.11 Obtained results on DBLP1, DBLP2 and PubMed datasets using `S-Kmeans`, `ITCC`, `VEM`, `PARAFAC`, `GSTC` and `VEM-T`. 42
2.12 Behavior of the γ_{kl} parameters at each iteration. 43
2.13 Comparaison of `CEM-T` and `VEM-T` in terms of time complexity and performances. 44

3.1 Difference between PLBM and SPLBM paramertization. 48
3.2 Goal of co-clustering of multiple graphs. 51
3.3 Political blogs dataset: Clustering with `PSBM` and `DC-SBM/Croinfo`. . . . 52
3.4 Graphical models: z_i is the label of row i, w_j is the label of column j. . . . 53
3.5 Political blogs dataset: Comparison of `PSBM`, `PLBM`, and `SPLBM` in terms of accuracy. 53
3.6 Amazon-products-10 dataset. 57
3.7 Comparison in terms of Accuracy and NMI for all datasets with `PSBM`, `PLBM`, `SPLBM` and `TSPLBM`. 58

3.8 CA applied on topic-tags matrix. . . . 58
3.9 Topic-tags frequencies matrix using top CA contributed tags. . . . 59
3.10 Co-tags graph of Nus-Wide-8. . . . 60
3.11 Normalized Log-likelihood *vs* NMI and ACC for all runs. . . . 60
3.12 Consensus clustering. . . . 61
3.13 Graphs clustering similarity. . . . 61
3.14 Comparison approach. . . . 62
3.15 Consensus based NMI comparison. . . . 63

4.1 General VEM-LBRM algorithm operation. . . . 66
4.2 Recommendation techniques. . . . 67
4.3 Data representation for proposed model. . . . 69
4.4 Graphical models: left mixture model without regression model(LBM), right proposed model(LBRM). . . . 69
4.5 Synthetic data: True regression plans according to the chosen parameters. . . . 72
4.6 Synthetic data: True co-clustering according to the chosen parameters. . . . 73
4.7 Obtained results on Strawberry dataset. . . . 75
4.8 Representation of mean of co-clusters. . . . 76
4.9 Co-clustering results on rating matrix obtained by VEM-LBRM on all datasets. 78
4.10 Precision of k-top recommendations using NMF, co-clustering and VEM-LBRM for all datasets. . . . 80
4.11 Recall of k-top recommendations using NMF, co-clustering and VEM-LBRM for all datasets. . . . 80

5.1 Waste collection and containers recommendation approach. . . . 85
5.2 Application interface for the proposed recommender system. . . . 87
5.3 Steps for waste collection approach. . . . 88
5.4 Reorganization of adjacency matrices of stores. . . . 88
5.5 Example of store clustering result. . . . 89
5.6 Obtained path collections by all algorithms for simulation 3. . . . 91
5.7 Application interface for the proposed waste colletion approach. . . . 92
5.8 Markdown analysis: Obtained co-clustering on each slice of tensor using VEM-T. . . . 93
5.9 Co-cluster analysis of markdowns using VEM-T. . . . 93
5.10 Different data sources for the EGC challenge. . . . 94
5.11 Reorganization of adjacency matrices of papers. . . . 96
5.12 Topics description according to terms. . . . 97
5.13 Description of topics according to authors. . . . 97
5.14 Frequencies of the most contributed terms to CA. . . . 98
5.15 Evolution of topics over time. . . . 98
5.16 Reorganization of adjacency matrices of author's graphs. . . . 99
5.17 Description of author's communities by affiliation. . . . 99
5.18 Male-female proportion by community of authors. . . . 100
5.19 Recommendation system for reviewing research papers. . . . 101

List of Tables

2.1 Expression of $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega})$ according various TLBM. 32
2.2 Evaluation of co-clustering in terms of NMI for binary datasets. 37
2.3 Evaluation of co-clustering in terms of NMI for continuous datasets. 37
2.4 Characteristics of datasets. 38
2.5 Evaluation of K-means, EMGMM, VEM and VEM-T in terms of NMI, ARI and ACC. 41

3.1 Characteristics of datasets. 55

4.1 Parameters generation for synthetic data. 73
4.2 (co)-clustering and prediction: "mean" and "standard deviation" in parentheses. 74
4.3 Description of Datasets. 77
4.4 Covariate impact on Datesets using the proposed VEM-LBRM algorithm. . . . 79
4.5 F-measure of k-top recommendations using NMF, co-clustering and VEM-LBRM for all datasets. 81

5.1 Results of VEM-LBRM recommendations on three real datasets. 86
5.2 Datasets description. 90
5.3 Evaluation and comparison of proposed approach for all datasets. 91
5.4 Examples of obtained recommendations. 101

Notation

$\mathcal{X}$	Tensor with size $n \times d \times v$
$\mathbf{X}$	Matrix with size $n \times d$
$\mathbf{X}^b$	Slice matrix b of tensor
x_{ij}	Matrix entry
$\mathbf{x}_{ij}$	Tensor vector entry with size v
$x_{i.}$	Degree of row i, where $x_{i.} = \sum_{j=1}^{d} x_{ij}$
$x_{.j}$	Degree of column j, where $x_{.j} = \sum_{i=1}^{n} x_{ij}$
x_{ij}^b	Tensor entry, equivalent to x_{ijb}
n	Size of first mode
d	Size of second mode
v	Size of third mode
g	Number of row clusters
m	Number of column clusters
$\mathbf{Z}$	Row partition matrix of size $n \times g$
$z_{.k}$	Cardinality of row cluster k, where $z_{.k} = \sum_{i=1}^{n} z_{ik}$
$\mathbf{W}$	Column partition matrix of size $d \times m$
$w_{.\ell}$	Cardinality of column cluster ℓ, where $w_{.\ell} = \sum_{j=1}^{d} w_{j\ell}$
π_k	Proportion of row cluster k
ρ_ℓ	Proportion of column cluster ℓ
$\boldsymbol{\beta}_{k\ell}$	Regression coefficient of block $k\ell$ with size $(v+1)$
$\sigma_{k\ell}$	Variance of regression error for the block $k\ell$
$\boldsymbol{\mu}_{k\ell}$	Mean of covariates per block $k\ell$
$\mu_{k\ell}^b$	Mean of covariates b per block $k\ell$
$\boldsymbol{\Sigma}_{k\ell}$	Co-variance matrix of covariates per block $k\ell$
$\boldsymbol{\lambda}_{k\ell}$	Parameters of distribution per block $k\ell$
$\lambda_{k\ell}^b$	Parameters of distribution per block $k\ell$ and slice b
$\gamma_{k\ell}^b$	The effect parameter of Poisson distribution for the block $k\ell$ and slice b
$\boldsymbol{\gamma}^b$	The effect parameter of Poisson distribution for the non-diagonal blocks in slice b
$\boldsymbol{\Omega}$	Model parameters
Φ	Probability density function (pdf)

Dedicated to my parents

Introduction

Today, the amount of collected data in different fields such as social networks, online shopping, and also the medical field grows exponentially. Several machine learning methods are developed to solve various problems related to this considerable data quantity. Supervised and unsupervised learning methods are essential for data analysis, prediction, and decision making in many areas, including electric consumption, medical image, and handwriting recognition. Even if supervised machine learning methods are the most popular, they depend, nevertheless, on the target labels, which must be known for the training dataset. However, in many problems in data science, the target labels are unknown. Therefore, the unsupervised machine learning (or clustering) paradigm is an indispensable tool for data mining. Clustering allows regrouping together similar objects into meaningful clusters, providing a summarization of data. Clustering is used for different data mining applications such as community detection, event detection, identifying fake news, text mining, pattern recognition, and recommendation systems.

To go further, Co-clustering, which can be viewed as an extension of clustering [Hartigan, 1972, Bock, 1979, Govaert, 1983] leads to reorganize a data matrix into homogeneous blocks (after appropriate permutations). Co-clustering plays an important role in a wide variety of applications where the data are generally organized in double-entry tables [Govaert and Nadif, 2013]. However in various situations, data can be reorganized into (3D) tensor and requires, therefore, appropriate clustering or co-clustering methods. This has driven many researchers to investigate new co-clustering models to consider tensor structures. To this end, two main strategies can be adopted;

- Adapt and restructure the tensor data to meet the requirements of existing methods. The restructuring of tensors, most frequently in a 2D matrices, causes a loss of information related to the ignoring of tensorial structures.

- Or design new methods fitting with the data structure. This strategy makes it possible to benefit from the tensor structure by exploiting the interdependence between the different slices and modes of tensor data.

However, most existing works are based on a tensor matrix decomposition [Banerjee et al., 2005, Wu et al., 2016, Feizi et al., 2017] and do not use tensor (co)-clustering under a probabilistic approach. Inspired by the famous statement made by Jean Paul Benzécri "The model must follow the data and not vice versa", we have chosen in this thesis to rely on appropriate mixture models. Thus, the contributions consist in adapting co-clustering based on the latent block model or (LBM, Latent Block Model), proposed for the first time in [Govaert and Nadif, 2003], to tensor data.

Motivation

In many areas, we are faced with dyadic data related to distinct entities like user-movie, document-word, individual-gene, and so on. For this kind of data, co-clustering proved its effectiveness in improving clustering results and also helping on the interpretation of the obtained results. Moreover, we can consider the co-clustering from tensor data linking more than two entities. Indeed, there are several data structured naturally as tensor data. Below, we present three applications (figure 1) using 3-way tensor structure, which will be detailed in the next chapters:

Recommender systems. When we consider movie recommendation, we can construct the following 3-way tensor Users $\times$ Movies $\times$ Covariates. These covariates can be, for instance, the age of users, the user's occupation, and movie genres. Using this tensor, the goal can be finding clusters of users interested in some movie genres (cf. chapter 2 ,4).

Documents categorization. Dealing with the document categorization problem, the majority of researches relies on the documents-terms matrix. However, we can use other available information concerning, for example, the authors or the citations. Thus, considering multiple relationships (co-terms, co-authors, co-references) between documents, leads to Documents $\times$ Documents $\times$ Relationships tensor. Then, the objective consists in finding groups of documents described by close relationships (cf. chapter 2,3).

Consumption profiles prediction. Approaching the problem of multivariate time series clustering, we can construct the tensor Consumers $\times$ Features $\times$ Time. The objective is to find some consumer groups having the same consumption patterns according to the time.

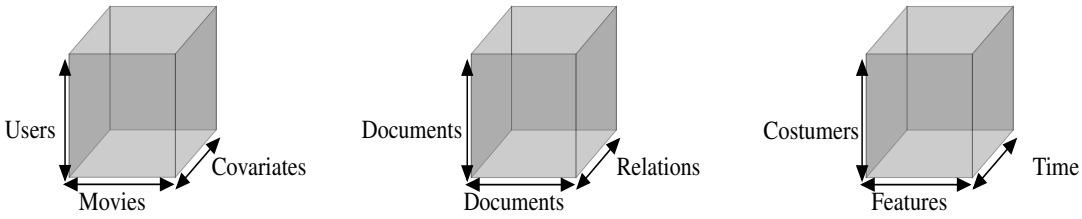


FIGURE 1: Examples of tensor data representation.

In this work, the tensor data structure is the main driver. Two issues can be modeled by tensor structure, namely, Multi-view clustering and ensemble clustering (or consensus) of multiple graphs. Indeed, these two issues can be represented by a three-way tensor, where each slice of tensor represents a view for the Multi-view methods and a graph for the consensus of multiple graphs methods, respectively. In this context, tensor-based methods are severely challenged by:

- **High dimensionality:** Today, we are able to collect a large number of features related to different objects yielding naturally to a tensor structure. The number of objects and features can be huge, and memory complexity can be increased very considerably. For instance a three-way tensor with size $10 \times 10 \times 20$ leads to fill 2000 cells, and a three-way tensor with size $1000 \times 1000 \times 2000$ to 2 billion cells.

- **Sparsity:** The high dimensionality problem is directly associated with the sparsity problem; the constructed tensors are often sparse. For instance, considering the review analysis in a recommender systems context, the reviews given by users about products can be represented by sparse tensor (around 99% of zeros).
- **Heterogeneity:** For high-dimensional data, some features can be irrelevant for a given classification due to outliers or corrupted data. Consequently, identifying these sets of irrelevant features and weighing their contribution to the clustering results appropriately, is an important issue. Heterogeneity is a concept used to describe this situation.

Contributions

The following is a brief description of all contributions in terms of three-way tensor data modeling in both contexts, unsupervised and supervised learning.

Tensor Latent Block Model for Co-clustering [Boutalbi et al., 2019a]. In our first contribution, we rely on the latent block model (LBM) [Govaert and Nadif, 2003], which is flexible in allowing us to model different types of data matrices. We extend its use to the case of tensor data in proposing a *Tensor LBM* (TLBM) taking into account different relations between entities. To show the interest of TLBM, we consider continuous, binary, and contingency tables datasets. To estimate the parameters, we develop a variational EM algorithm `VEM-T`. Its performances are evaluated on synthetic and real datasets to highlight different possible applications.

Sparse Tensor Co-clustering [Boutalbi et al., 2019b]. This contribution consists in extending the use of the *Sparse Poisson Latent Block Model* (SPLBM) [Ailem et al., 2017]. To tackle the document clustering problem from sparse tensor data obtained from a set of documents, we propose a Tensor SPLBM (TSPLBM) which is parsimonious and tailored for this kind of data. Then, we propose a suitable tensor co-clustering algorithm `TSPLBM`. Empirical results on several real-world text datasets highlight the advantages of our proposal which improves the clustering results of documents.

Implicit Consensus Clustering from Multiple Graphs [Boutalbi et al., 2020]. Dealing with relational learning generally relies on tools modeling relational data. An undirected graph can represent these data with vertices depicting entities and edges describing the relationships between the entities. These relationships can be well represented by multiple undirected graphs over the same set of vertices, with edges arising from different graphs catching heterogeneous relations. The vertices of those networks are often structured in unknown clusters with varying properties of connectivity. These multiple graphs can be structured as a three-way tensor, where each slice of tensor depicts a graph which is represented by a count data matrix. To extract relevant clusters, we propose an appropriate model-based co-clustering capable of dealing with multiple graphs. The proposed model can be seen as a suitable tensor extension of mixture models of graphs, while the obtained co-clustering can be treated as a consensus clustering of nodes from multiple graphs. Applications on real datasets show the interest of our contribution.

Model-based co-clustering via latent block regression model [Boutalbi et al., 2018]. Clusterwise methods aim to obtain a partition simultaneously into g clusters and g local models optimizing a given criterion. This objective is useful in many field domains, such as recommender systems. However, when dealing with high dimensional sparse data, such as in recommender systems, co-clustering turns out to be more beneficial than one-sided clustering even if one is interested in clustering along one dimension only. Thereby, co-clusterwise is a natural extension of clusterwise. Unfortunately, all of the existing approaches do not take into account covariates on both dimensions of a data matrix. This contribution consists in proposing a Latent Block Regression Model (LBRM) overcoming this limit. To fit LBRM, we propose a new algorithm performing simultaneously co-clustering and regression where a linear regression model characterizes each block. Placing the estimate of the model parameters under the maximum likelihood approach, we derive a Variational EM (VEM) algorithm and propose to evalute results for recomender systems.

Overview

The rest of this thesis is organized into five chapters. The main contents of each chapter are summarized below :

Chapter 1. In this chapter, we first describe the most popular clustering and co-clustering approaches. Then, we introduce the tensor data structure and some proprieties related to three-way tensors. Finally, we describe the state-of-the-art relevant to tensor analysis and clustering.

Chapter 2. This chapter is devoted to the co-clustering of tensor data. We first review LBM in detail for the data matrix. Then, we describe the proposed extension Tensor LBM (TLBM) and its corresponding algorithm `VEM-T`.

Chapter 3. In this chapter, we tackle the clustering problem of multiple graphs. We first introduce the *Stochastic Bock Model* (SBM) [Karrer and Newman, 2011] and show the connection between SBM and LBM models. After that, we present an appropriate novel extension of LBM for clustering of multiple graphs. Moreover, we demonstrate the advantages of implicit consensus obtained by the proposed `TSPLBM` algorithm comparing to traditional consensus methods.

Chapter 4. In this chapter, we address the problem of simultaneous supervised learning and co-clustering. We detail the proposed *Latent Block Regression Model* (LBRM) and establish some connections with classical existing mixture models. Furthermore, we derive the `VEM-LBRM` algorithm and show their effectiveness on recommender systems application.

Chapter 5. In this chapter, we propose to apply our algorithms for original applications. First, we address the issue of waste management. The objective is to improve and optimize different tasks of waste management. The second application concerns the *EGC conference challenge*. We will detail all analyses and present obtained results.

Publications

Accepted papers

- Rafika Boutalbi, Lazhar Laboid and Mohamed Nadif, “Tensor Latent Block Model for Co-clustering”, International Journal of Data Science and Analytics (IJDSA), Springer, 2020.
- Rafika Boutalbi, Lazhar Laboid and Mohamed Nadif, “Défi EGC 2020 : Analyse tensorielle de données issues de la conférence EGC”, EGC, 2020, Brussels, Belgium.
- Rafika Boutalbi, Lazhar Laboid and Mohamed Nadif, “Classification croisée de données tensorielles”, SFC conference (Société Francophone de Classification), pp 95-98, 2019, Nancy, France.
- Rafika Boutalbi, Lazhar Labiod and Modamed Nadif, “Sparse tensor co-clustering as a tool for document clustering”, pp 1157–1160, SIGIR, 2019, Paris, France.
- Rafika Boutalbi, Lazhar Laboid and Mohamed Nadif, “Co-clustering from Tensor Data”, The 23rd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp 370-383, 2019, Macau, China.
- Rafika Boutalbi, Lazhar Laboid and Mohamed Nadif, “Classification croisée et regression locale”, SFC conference (Société Francophone de Classification), pp 87-90, 2018, Paris, France.

Submitted papers

- Rafika Boutalbi, Lazhar Laboid and Mohamed Nadif, “Implicit Consensus Clustering from Multiple Graphs”.

Chapter 1

(Co)-clustering of two-way and three-way tensor data

In this chapter, we present an overview of the most popular clustering and co-clustering of two-way and three-way tensor data methods. First, we will briefly present the concept of clustering and the most known clustering approaches. Second, we will provide a definition of the concept of co-clustering, highlight the main advantages of co-clustering compared to one-way clustering, and present some popular co-clustering approaches. Finally, we will give a brief survey about three-way tensor data; the objective here is not to detail all existing approaches but to give an outline of the significant definitions, proprieties, and methods in this area.

1.1 Clustering

Clustering, sicks to group together a set of data points (objects) into *homogeneous clusters or natural classes*, in a way which ensures that objects within a cluster are similar to each other. Thereby, the basic problem of clustering, or unsupervised learning methods, can be summarized as follows "Given a set of objects, partition them into a set of clusters which are as similar as possible". Several clustering approaches have been developed and applied in many fields, such as text mining, bio-medical, event detection, etc. Clustering is an important tool for data analysis and data mining. In the clustering problem, the data is represented by a set of n objects described by d variables $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ where $\mathbf{x}_i = \{x_{i1}, x_{i2}, ..., x_{id}\}$. Hence, the data can be represented by a matrix $\mathbf{X}$ with size $n \times d$. The objective is to group objects into homogeneous clusters based on distance measure or similarity by optimizing an objective function leading to a variety of clustering approaches. Hereafter we present a brief description of the most popular clustering methods.

1.1.1 Hierarchical clustering

Hierarchical methods aim to provide multiple clustering levels. The discovered hierarchy is constructed based on a given distance (Euclidean, Ward, etc.) and can be represented by a dendrogram (see figure 1.1). There are two principal hierarchical clustering approaches: *Agglomerative* and *Divisive* methods [Everitt et al., 2011, Murtagh and Contreras, 2017].

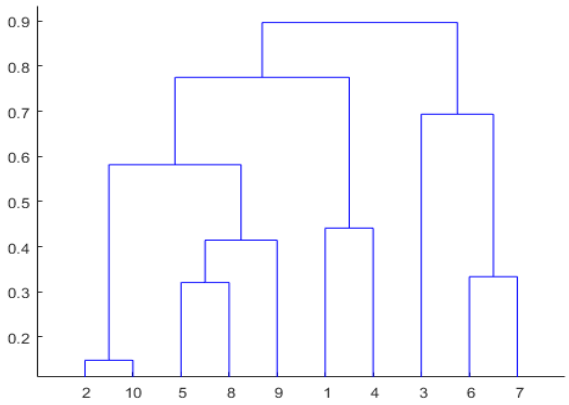


FIGURE 1.1: An hierarchical clustering represented by a dendrogram.

Agglomerative methods

These approaches, also known as *Bottom-up*, put each object in a distinct cluster; after that, the closest clusters are merged. This merging step requires to choose a distance to measure the dissimilarity between clusters, and linkage criteria. There are several linkage criteria, such as single-linkage [Sibson, 1973], complete-linkage [Legendre and Legendre, 1998], average-linkage [Sokal et al., 1958], and Ward-linkage [Ward, 1963]. The agglomerative approaches are not limited to the methods mentioned above. Many other methods have been proposed in the literature. For more details, the reader can refer to survey papers [Everitt et al., 2011, Murtagh and Contreras, 2017].

Divisive methods

On the other hand, divisive approaches, also denoted as *Top-Down*, assume that all objects are in the same cluster. The splitting step (opposite to merging step in agglomerative methods) are used to obtain smaller clusters until a stopping condition is reached. MONA and DIANA [Kaufman and Rousseeuw, 1990] are two divisive approaches. These methods are less commonly used due to their computational complexity. Some other techniques can be found in [Everitt et al., 2011].

The main advantage of hierarchical clustering is that it does not require the number of clusters as an input. The cluster's number is obtained by the cutting method. However, the hierarchical approaches suffer from high time complexity and are not suitable for large datasets.

1.1.2 Density-based approaches

This variety of algorithms assumes that clusters have different density. In contrast with many popular clustering methods, this type of algorithm allows us to discover clusters with various volumes. One of the most known density-based approaches is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [Ester et al., 1996]. The DBSCAN algorithm

requires two parameters, (i) ε: which represents a threshold to decide if two points are neighbors or not, and (ii) $minPoints$: the minimum number of points to form a dense region. DBSCAN iterates over the points of the dataset. For each point, it detects all the points that can be reached by density from this point based on the epsilon threshold ε. If this neighborhood has more than $minPoints$ points, the same operation is applied, and so on, until they can not expand the cluster. If the point considered is not a core point, i.e., it does not have enough neighbors, it will be labeled as noise. This allows DBSCAN to be robust to outliers since this mechanism isolates them. However, the algorithm is sensitive to the settings of parameters ε and $minPoints$. To overcome this drawback, OPTICS [Ankerst et al., 1999] is proposed; it is a generalization of DBSCAN and does not require parameters. For more details and developments in density-based approaches, the reader can refer to [Kriegel et al., 2011a,b].

1.1.3 Graph-based approaches

Community detection is becoming increasingly significant since graph representation is used in several applications such as, social media, web mining, bio-medical.Graph clustering aims to discover g communities (or clusters) into graphs (see figure 1.2). A graph is formed by a set of vertices (or nodes) connected by a set of edges. The objective is to regroup the highly connected nodes in the same cluster, thus maximizing the number of edges inside each cluster and minimizing the number of edges between communities. Several formulations of graph

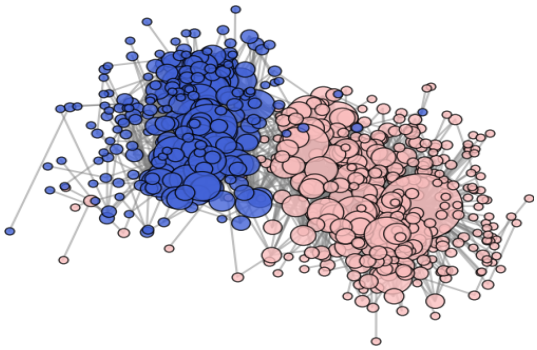


FIGURE 1.2: Political blogs communities.

clustering problem have been proposed. The most direct way to identify a partition in a graph is to solve the *minimum cut (mincut)* problem [von Luxburg, 2007]. Multiple varieties of the *minimum cut* problem have been proposed. For instance, the *minimum ratio cut* problem [Hagen and Kahng, 1992] introduces a division by the size of each cluster into a *mincut* objective function. This overcomes the problem of separating one node from the rest of the graph. In the same way, the *minimum normalized cut* problem [Jianbo Shi and Malik, 2000] introduces the division by the sum of node degrees within each cluster. In the same topic, [Ding et al., 2001] proposed the *min-max cut* problem, which consists of both minimizing the density of inter-cluster edges and maximizing the density of intra-cluster edges.

Spectral clustering (SC) is another way to deal with graph data. Due to their simplicity, mathematical elegance, and efficiency, *SC* has attracted the interest of researchers and has been applied in many fields. The principal of *SC* is to find clustering from eigenvectors of

the Laplacian matrix of the graph. For more details about *Spectral clustering*, please refers to [Jianbo Shi and Malik, 2000, von Luxburg, 2007].

Recently, [Newman and Girvan, 2004] proposed a graph clustering approach based on *Modularity* measure. The modularity aims to maximize the difference between intra-cluster edges and the expectation of this value in a random graph. In [Blondel et al., 2008], the authors proposed the *Louvain* algorithm, also based on *Modularity* measure but using a hierarchical strategy to construct the communities.

1.1.4 Partitional clustering approaches

Partitional clustering aims to find g groups of similar objects based on some features. Unlike hierarchical clustering, which discovers structures with the hierarchical relationships, partitional clustering discovers disjoint clusters. The advantages of partitional clustering are its simplicity and scalability.

Centroid-based approaches

The centroid-based approaches are very intuitive and essential tools for data clustering. In fact, these approaches assume that each cluster is represented by a *centroid*, which is not necessarily an object from a set of observations. The objective is partitioning objects into g groups by optimizing an objective function. The obtained results are the cluster of each object and prototypical object (*centroid*) designated as a representative for each cluster. The multiple ways of choosing the centroid and the objective function give rise to many centroid-based methods.

One of the most popular centroid-based approaches is the well-known *k-means* algorithm [MacQueen, 1967, Bock, 2007]. The k-means aims to find g clusters by minimizing the Euclidean distance between each object and its cluster centroid. The objective function minimized by the *k-means* algorithm can be written as follows :

$$\sum_{i=1}^{n}\sum_{k=1}^{g} z_{ik}\|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2,$$

where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the ith observation, $\boldsymbol{\mu}_k$ is the centroid of the cluster k, z_{ik} equals to 1 if the ith observation belongs to cluster k, and 0, otherwise. To optimize the objective function, the *k-means* alternate two following steps until convergence :

- **Initialization**: Select g centroids randomly from the set of objects.
- **Optimization**: Alternate the two following steps
  - Assignment of objects to the clusters based on euclidean distance between the object and the centroids.
  - Updating the centroid based on the new assignment (mean aggregation).

In this case, the convergence is achieved when the objective function becomes stationary or quasi-stationary.

Mixture-based approaches

The mixture model is one of the most important and powerful clustering approaches. It was introduced by [Pearson, 1911] and can be dealt with different types of data (continuous, binary, and contingency table) and models various cluster's shape. Mixture models assume that a set of objects is composed of g sub-sets characterized by probability distributions. Thus the data matrix $\mathbf{X} = [\mathbf{x}_i] \in \mathbb{R}^{n \times d}$ is assumed to be an independent and identically distributed (i.i.d.) sample $\mathbf{x}_i$ where $\mathbf{x}_i \in \mathbb{R}^d$ is generated according to a probability density function:

$$f(\mathbf{x}_i, \mathbf{\Omega}) = \sum_{k=1}^{g} \pi_k \Phi(\mathbf{x}_i, \boldsymbol{\lambda}_k),$$

and considering all observed data :

$$f(\mathbf{X}, \mathbf{\Omega}) = \prod_{i=1}^{n} \sum_{k=1}^{g} \pi_k \Phi(\mathbf{x}_i, \boldsymbol{\lambda}_k),$$

Subject to constraints :

$$\forall k = 1, ..., g, \pi_k \in]0,1[, \; and \sum_{k=1}^{g} \pi_k = 1,$$

where π_k represents the proportion of each cluster, $\Phi(\mathbf{x}_i, \boldsymbol{\lambda}_k)$ is the density of the observation $\mathbf{x}_i$ from the kth component. And $\boldsymbol{\lambda}_k$ is a vector of parameters depending on the selected density function. For instance, considering a Gaussian distribution, the set of parameters for each component is $\boldsymbol{\lambda}_k = \{\boldsymbol{\mu}_k, \mathbf{\Sigma}_k\}$. Figure 1.3 shows a mixture of two Gaussian distributions. To estimate the parameters $\mathbf{\Omega}$, given the observed data $\mathbf{X}$ in this context of mixtures

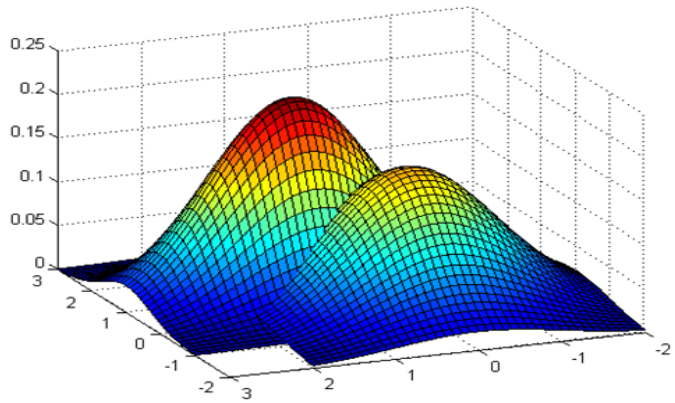


FIGURE 1.3: Mixture of two Gaussian density functions.

of a probability distribution, two popular approaches were proposed in the literature, namely the Maximum Likelihood (ML), and Classification ML (CML) [Scott and Symons, 1971, Symons, 1981]. The Expectation-Maximization (EM) algorithm can be used to estimate the model parameters maximizing the ML. The Classification EM (CEM) algorithm is a variant

of EM maximizing CML objective function yielding to a soft clustering [Celeux and Govaert, 1992]. The objective of these two approaches is finding the parameters maximizing the likelihood of the observed data $\mathbf{X}$. Both techniques rely on the complete data log-likelihood because it is hard to work directly with the likelihood function, given as follows:

$$L_C(\mathbf{\Omega}, \mathbf{X}, \mathbf{Z}) = \sum_i \sum_k z_{ik} log(\pi_k) + \sum_i \sum_k z_{ik} \log \Phi(\mathbf{x}_i, \boldsymbol{\lambda}_k).$$

The advantages of the mixture models approach are its flexibility and adaptation with various situations, including the presence of heterogeneous data and outliers. Moreover, its associated estimators of posterior probabilities can result in both fuzzy and/or hard clustering using the maximum a posteriori (MAP) principal.

1.1.5 Clustering evaluation metrics

Evaluating clustering results is not a trivial task. To this end, we can use benchmark datasets with a true partition. The objective is comparing the true partition with the clustering partition obtained by clustering algorithms. Many measures are available, and the most popular is the accuracy, which corresponds to the percent of correct predictions. However, the clustering accuracy is not always a reliable measure when the clusters are not balanced, and the number of clusters is high. To better appreciate the quality of our clustering approach, in this thesis, we retain two widely used measures to assess the quality of clustering, namely the Normalized Mutual Information and the Adjusted Rand Index.

Accuracy. The clustering accuracy noted (Acc) discovers the one-to-one relationship between two partitions and measures the extent to which each cluster contains data points from the corresponding class. It is defined as follows:

$$\text{Acc} = \frac{1}{n} \sum_{i=1}^{n} \delta(\mathcal{C}_i, map(\mathcal{P}_i))$$

where n is the total number of samples, $\mathcal{P}_i$ is the i^{th} obtained cluster and $\mathcal{C}_i$ is the true i^{th} class provided by the data set. $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(\mathcal{P}_i)$ is the permutation mapping function that maps the obtained label $\mathcal{P}_i$ to the equivalent label from the data set. The best mapping can be found by using the Kuhn-Munkres algorithm [Munkres, 1957, Bourgeois and Lassalle, 1971].

Normalized Mutual Information (NMI) The NMI [Strehl and Ghosh, 2002] measure is estimated by

$$\text{NMI} = \frac{\sum_{k,\ell} \frac{n_{k\ell}}{n} \log \frac{n n_{k\ell}}{n_k \hat{n}_\ell}}{\sqrt{(\sum_k n_k \log \frac{n_k}{n})(\sum_\ell \hat{n}_\ell \log \frac{\hat{n}_\ell}{n})}},$$

where n_k denotes the number of data contained in cluster $\mathcal{C}_k (1 \leq k \leq K)$, $\hat{n}_\ell$ is the number of data belonging to the class $\mathcal{L}_\ell (1 \leq \ell \leq K)$, and $n_{k\ell}$ denotes the number of data that are in the intersection between cluster $\mathcal{C}_k$ and class $\mathcal{L}_\ell$. Intuitively, NMI quantifies how much the estimated clustering is informative about the true clustering.

Ajusted Rand Index (ARI) The ARI [Liu et al., 2013b] measure quantifies the similarity between two data clustering partitions. From a mathematical standpoint, the Rand index is related to the accuracy. The adjusted form of the Rand Index is:

$$\text{ARI} = \frac{\sum_{k,\ell}\binom{n_{k\ell}}{2} - \left[\sum_k \binom{n_k}{2}\sum_\ell \binom{\hat{n}_\ell}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_k \binom{n_k}{2} + \sum_\ell \binom{\hat{n}_\ell}{2}\right] - \left[\sum_k \binom{n_k}{2}\sum_\ell \binom{\hat{n}_\ell}{2}\right] / \binom{n}{2}}.$$

The ARI is related to the clustering accuracy and measures the degree of agreement between an estimated clustering and a reference clustering. All of Acc, NMI and ARI are equal to 1 if the resulting clustering is identical to the true one.

1.2 Co-clustering

Unlike clustering approaches which reorganize only rows (objects) of data matrix, co-clustering (or bi-clustering) is a set of methods for simultaneous clustering of rows (objects, individuals, instances) and columns (features, objects) into meaningful co-clusters linked row clusters and columns clusters[Bock, 1979]. It aims to discover homogeneous blocks, provide an improved results and an easier interpretation of obtained results, especially for sparse data (see figure 1.4). The co-clustering proved their effectiveness on many applications such as text mining, micro-array analysis, image clustering.

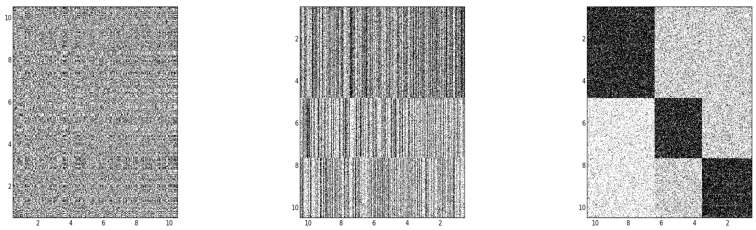


FIGURE 1.4: (left) Original matrix, (middle) Rows clustering results, and (right) Co-clustering results.

1.2.1 Metric-based approaches

Metric-based approaches are inspired by *centroid-based clustering*. The intuition is that a scalar $\mu_{k\ell}$ can summarize each co-cluster of the data matrix. For this end, metric based co-clustering methods consist of the optimization of the following objective function :

$$\sum_{i,j,k,\ell} z_{ik} w_{jl} (x_{ij} - \mu_{k\ell})^2,$$

where x_{ij} is an entry of the data matrix, $\mathbf{Z}$ and $\mathbf{W}$ represent the partition matrices of rows and columns, respectively, and $\boldsymbol{\mu} = (\mu_{k\ell})$ summarizing the original matrix. Metric-based methods consist of minimization of the difference between the original matrix and the summarized one using co-clustering.

The above optimization problem is intractable — nevertheless, an optimal solution by using the double k-means algorithm has been developed in [Govaert, 1995]. Multiple metric-based co-clustering algorithms using this principle has been proposed. We can site, `CROEUC`, `CROBIN`, and `CROKI2`, for continuous, binary, and contingency tables, respectively [Govaert, 1983].

1.2.2 Graph-based approaches

Considering a bipartite graph, graph-based approaches aim to clusters the set of nodes on g groups, such as the sum of weights of edges between clusters is maximized, and the sum of the weight of the edges within clusters is minimized. Dhillon [2001] proposed a Spectral Co-clustering method (SpecCo), which consists of partitioning a bipartite graph minimizing the cut objective function. Different algorithms based on graph modularity optimization have been developed in [Labiod and Nadif, 2011] and more recently in [Ailem et al., 2015, Role et al., 2019].

1.2.3 Matrix factorization-based approaches

Matrix factorization approaches have demonstrated an interest in a variety of fields. Moreover, several works have explored the connexion between Non-Negative Matrix factorization (NMF) and co-clustering. Even if co-clustering is not the main purpose of NMF, it can be used to perform this task [Ding et al., 2006, Hosseini-Asl and Zurada, 2014]. The Non-Negative Matrix Tri-Factorization (NMTF) method has been further developed to address various aspects of co-clustering, including high dimensionality; see for instance [Wang et al., 2011, Allab et al., 2016, 2017, Salah et al., 2018]. Given a positive data matrix $\mathbf{X}$, NMTF decomposes $\mathbf{X}$ on three factors $\mathbf{Z}$, $\mathbf{S}$, and $\mathbf{W}$ by optimizing the following objective function:

$$\min_{\mathbf{Z}\geqslant 0,\mathbf{S}\geqslant 0,\mathbf{W}\geqslant 0} \|\mathbf{X}-\mathbf{Z}\mathbf{S}\mathbf{W}^T\|,$$

where $\|.\|$ is the Frobenius norm, $\mathbf{Z}$ and $\mathbf{W}$ are two positive matrices, which can be converted to membership matrix of rows and columns, and $\mathbf{S}$ is also a positive matrix summarizing $\mathbf{X}$ considering the co-clustering.

To go further, there are other NMF-based approaches including supplementary constraints on the matrices; we can cite, for instance, non-negative block value decomposition (NBVD) [Long et al., 2005], and orthogonal three factors NMF (ONM3F and ONMTF) [Ding et al., 2006, Yoo and Choi, 2010]. For more details about NMF variants; see for instance [Li and Ding, 2018].

1.2.4 Model-based approaches

Model-based co-clustering approaches are powerful techniques providing more flexibility, robustness, and allowing us to model different types of data. Moreover, the generative model-based approach offers theoretical foundations considering the metric-based methods. The Latent Block Model (LBM) is a popular model-based co-clustering approach [Govaert and Nadif, 2003, Nadif and Govaert, 2005, 2010, Govaert and Nadif, 2010, 2013, 2018]. It assumes that the data matrix can be split into co-clusters (or bi-clusters), and a univariate probability distribution function describes each of the co-cluster. We will see in our propositions

that this definition changes, and we might use a multivariate probability distribution function considering a co-clustering of tensor data.

The latent block model [Govaert and Nadif, 2013] in $g \times m$ blocks is defined as follows. Given a matrix $\mathbf{X}$ of size $n \times d$, we assume that there is a couple of partitions $(\mathbf{z}, \mathbf{w})$ where $\mathbf{z}$ is partitioned in g clusters on the set of rows I and $\mathbf{w}$ is partitioned in m clusters on the set of columns J, such that each element x_{ij} belonging to the block $k\ell$ is generated according to a probability distribution, where k represents the class of row i, while ℓ represents the class of column j. The $\mathbf{z}$ partition can be represented by a vector of labels or by matrix $\mathbf{Z} = (z_{ik})$ of size $n \times g$ where $z_{ik} = 1$ if i belongs to the class k, and $z_{ik} = 0$ otherwise. In the same way, the $\mathbf{w}$ partition can be represented by a label vector or by a column classification matrix $\mathbf{W} = (w_{j\ell})$ of size $d \times m$ where $w_{j\ell} = 1$ if j belongs to the class ℓ, and $w_{j\ell} = 0$ otherwise. Under the independence assumption $p(\mathbf{Z}, \mathbf{W}) = p(\mathbf{Z})p(\mathbf{W})$ and noting $\mathcal{Z}$ and $\mathcal{W}$ the sets of all possible partitions $\mathbf{Z}$ and $\mathbf{W}$, the likelihood of the observed data $f(\mathbf{X}; \mathbf{\Omega})$ is given by:

$$\sum_{(\mathbf{z},\mathbf{w})\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,\ell} \rho_\ell^{w_{j\ell}} \prod_{i,j,k,\ell} \left(\Phi(x_{ij}; \boldsymbol{\lambda}_{k\ell})\right)^{z_{ik}w_{j\ell}}, \tag{1.1}$$

where $\mathbf{\Omega} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\lambda})$ are the unknown parameters of LBM 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 proportions of clusters, and $\boldsymbol{\lambda}_{k\ell}$ represents the parameters of $k\ell$ block distribution. The classification log-Likelihood takes the following form:

$$L_C(\mathbf{Z}, \mathbf{W}, \mathbf{\Omega}) = \sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell + \sum_{i,j,k,\ell} z_{ik} w_{j\ell} \log(\Phi(x_{ij}; \boldsymbol{\lambda}_{k\ell})). \tag{1.2}$$

The Latent Block Model will be detailed in section 2.2.1.

1.3 Clustering and analysis of Tensor data

In this section, we define a tensor data and the main properties of three-way tensors. We present a classification of the tensor-based model and briefly describe the principal tensor approaches. Finally, we review the most popular applications in this context.

1.3.1 Notation and Preliminaries

A tensor is a multidimensional array, which is also known as the N-way and Nth-order tensor. A tensor can be viewed as an element product of N vector spaces [Kolda and Bader, 2009]. This notion of tensors should not be confused with tensors in physics and mathematics fields such as stress and strain tensors [Frankel, 2012]. A third-order tensor has three dimensions and then three indices, as shown in Figure 1.5. A first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are called higher-order tensors.

Tensor representation

The notation used here is very close to that introduced by [Kiers, 2000] for third-order tensor. Notice that scalars are represented by lowercase letters e.g. x, vectors are expressed by a bold lowercase letter e.g. $\mathbf{x}$. The matrices are denoted by bold capital letters, e.g. $\mathbf{X}$, and tensors

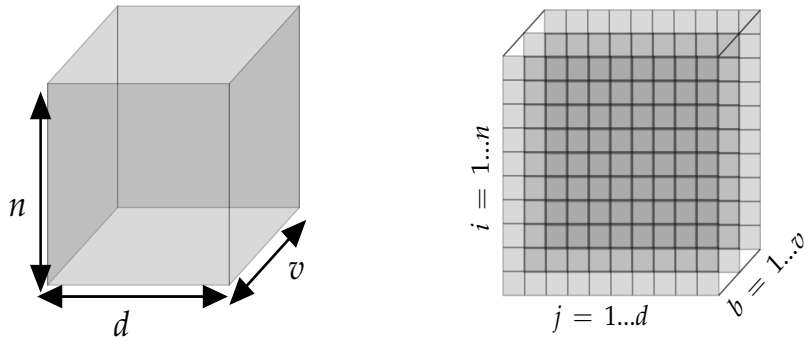
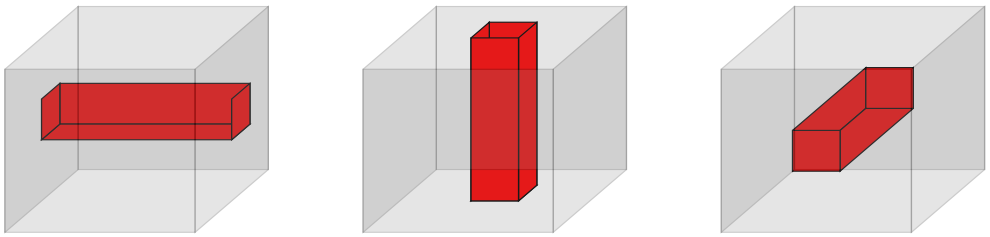


FIGURE 1.5: Third-way tensor data representation.

by bold capital Euler letters, e.g. $\mathcal{X}$. The ith element of $\mathbf{x}$ is denoted as x_i, the element (i, j) of $\mathbf{X}$ is expressed by $x_{i,j}$, and $x^b_{i,j}$(or $x_{i,j,b}$) represents the element (i, j, b) of a tensor.

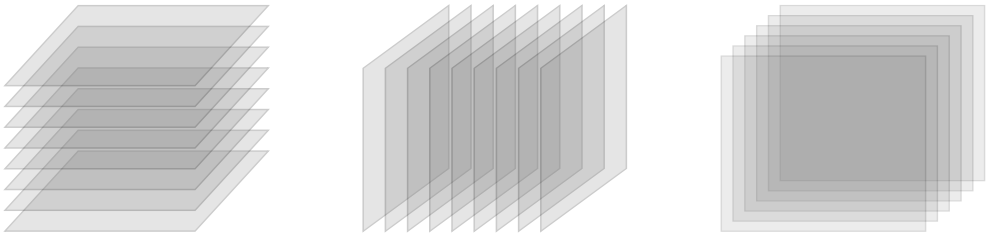
The order of tensor is referred to as the number of dimensions, also called ways or modes. Then one-mode tensor is a vector, second-order tensor is a matrix, and third-order tensor is a cuboid. In the following, for $\mathcal{X}$ tensor, we will denote the tensor entry $\mathbf{x}_{ij:}$ by $\mathbf{x}_{ij} = (x^1_{ij}, \ldots, x^b_{ij}, \ldots, x^v_{ij})$; then $x^b_{i.} = \sum_j x^b_{ij}$ and $x^b_{.j} = \sum_i x^b_{ij}$. (see figure 1.6).



(A) First-mode (Row) (B) Second-mode (Column) (C) Third-mode (Tube)

FIGURE 1.6: Different representations of tensor elements.

We can decompose tensors into slices. These slices differ according to the considered mode. Figure 1.7 shows Horizontal, Lateral, and Frontal slices of tensor denoted by $\mathbf{X}_{i::}$, $\mathbf{X}_{:j:}$, and $\mathbf{X}_{::b}$, respectively. The frontal slices can be expressed by $\mathbf{X}^b$.



(A) Horizontal slices. (B) Lateral slices. (C) Frontal slices.

FIGURE 1.7: Slices representations of tensor.

Tensor proprieties

Symmetric Tensor. Before introducing the definition of symmetric and semi-symmetric tensors, we define *Cuboid* and *Cube* tensor. A *cuboid* is a three-way tensor with size $n \times d \times v$, and a *cube* is also a three-way tensor with size $n \times n \times n$. A *cube* is *supersymmetric* if $x_{ijb} = x_{jib} = x_{ibj} = x_{jbi} = x_{bij} = x_{bji}$ for $\forall i, j, k = 1...n$.

Cuboid with size $\in \mathbb{R}^{n \times n \times v}$ and *cube* can be *semi-symmetric* in one mode. For instance, if we consider the third mode (frontal slices), the tensor is *semi-symmetric* if $\mathbf{X}^b = (\mathbf{X}^b)^T$ for $\forall b = 1...v$.

Diagonal Tensor. A diagonal of *cube* can be denoted by x_{iii}(or x^i_{ii}) for $\forall i, j, k = 1...n$. Figure 1.8 shows a *cube* with diagonal equals to one.

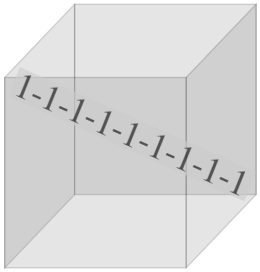


FIGURE 1.8: *Cube* tensor with ones on diagonal.

Tenor norm . The Frobenius norm of a three-way tensor $\mathcal{X} \in \mathbb{R}^{n \times d \times v}$ equals to the square root of the sum of its squared elements as:

$$\|\mathcal{X}\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{d} \sum_{b=1}^{v} x_{ijb}^2}.$$

Tensor products

Outer product. An outer product of three vectors $\mathbf{r} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^d$, and $\mathbf{s} \in \mathbb{R}^v$ can be represented as a third-way tensor $\mathcal{X} \in \mathbb{R}^{n \times d \times v}$:

$$\mathcal{X} = \mathbf{r} \circ \mathbf{c} \circ \mathbf{s},$$

where each element x_{ijb} equals to $r_i c_j s_b$

Dot product. In this part, we consider only the dot product, also known as the inner product between a three-way tensor and matrix (For more information about tensor products see e.g. [Bader and Kolda, 2006]). The multiplication between a tensor $\mathcal{X} \in \mathbb{R}^{n \times d \times v}$ and a matrix $\mathbf{M}$ must consider one of the three modes. Regarding the first mode with $\mathbf{M} \in \mathbb{R}^{m \times n}$, the obtained tensor with size $m \times d \times v$, can be computed as :

$$(\mathcal{X}\mathbf{M})_{ijb} = \sum_{a=1}^{n} x_{ajb} m_{ia}.$$

Hadamard product. A Hadamard product $\mathcal{H}$ between two third-mode tensors $\mathcal{X}$ and $\mathcal{Y} \in \mathbb{R}^{n\times d\times v}$ can be computed as element-wise product $h_{ijb} = x_{ijb} * y_{ijb} \forall i = 1...n, j = 1...d, b = 1...v$ [Kressner and Perisa, 2017]. The matrix form of Hadamard product can be written as:

$$\mathcal{H} = \mathcal{X} * \mathcal{Y}.$$

Kronecker product. A Kronecker product (also known tensor product) between two matrices $\mathbf{M} \in \mathbb{R}^{n\times d}$ and $\mathbf{Y} \in \mathbb{R}^{v\times m}$ is denoted by $\mathbf{M} \otimes \mathbf{Y}$. The obtained matrix with size $(nd) \times (vm)$ can be computed by:

$$\mathbf{M} \otimes \mathbf{Y} = \begin{bmatrix} m_{11}\mathbf{Y} & m_{12}\mathbf{Y} & \cdots & m_{1d}\mathbf{Y} \\ m_{21}\mathbf{Y} & m_{12}\mathbf{Y} & \cdots & m_{2d}\mathbf{Y} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1}\mathbf{Y} & m_{n2}\mathbf{Y} & \cdots & m_{nd}\mathbf{Y} \end{bmatrix}. \tag{1.3}$$

Transforming tensor to matrix

Tensor matricization. We can transform an N-way tensor into a matrix. This task is known as matricization, unfolding, or flattening. For instance, if we consider a three-way tensor $\mathcal{X}$ with size $3 \times 4 \times 2$ (see eq 1.4), we can rearrange it as 3×8, 4×6, or 2×12 depending on the selected mode.

$$\mathbf{X}^1 = \begin{bmatrix} 3 & 2 & 8 & 5 \\ 5 & 4 & 0 & 9 \\ 7 & 1 & 9 & 6 \end{bmatrix} \quad \mathbf{X}^2 = \begin{bmatrix} 10 & 12 & 11 & 15 \\ 14 & 17 & 10 & 19 \\ 18 & 13 & 14 & 16 \end{bmatrix}. \tag{1.4}$$

The following matrices represent the result of tensor matricization for each mode respectively :

$$\mathbf{X}_{mode1} = \begin{bmatrix} 3 & 2 & 8 & 5 & 10 & 12 & 11 & 15 \\ 5 & 4 & 0 & 9 & 14 & 17 & 10 & 19 \\ 7 & 1 & 9 & 6 & 18 & 13 & 14 & 16 \end{bmatrix} \quad \mathbf{X}_{mode2} = \begin{bmatrix} 3 & 5 & 7 & 10 & 14 & 18 \\ 2 & 4 & 1 & 12 & 17 & 13 \\ 8 & 0 & 9 & 11 & 10 & 14 \\ 5 & 9 & 6 & 15 & 19 & 16 \end{bmatrix}, \tag{1.5}$$

$$\mathbf{X}_{mode3} = \begin{bmatrix} 3 & 5 & 7 & 2 & 4 & 1 & 8 & 0 & 9 & 5 & 9 & 6 \\ 10 & 14 & 18 & 12 & 17 & 13 & 11 & 10 & 14 & 15 & 19 & 16 \end{bmatrix}. \tag{1.6}$$

Tensor compression. We can compress a three-way tensor $\mathcal{X}$ into matrices considering modes. For instance, a tensor with size $5 \times 4 \times 2$ can be transformed into matrices with size 4×2, 5×2, or 5×4 using some aggregation functions(sum, mean, median, etc.). Figure 1.9 shows the results of tensor compression according to the three modes 1, 2, and 3, respectively.

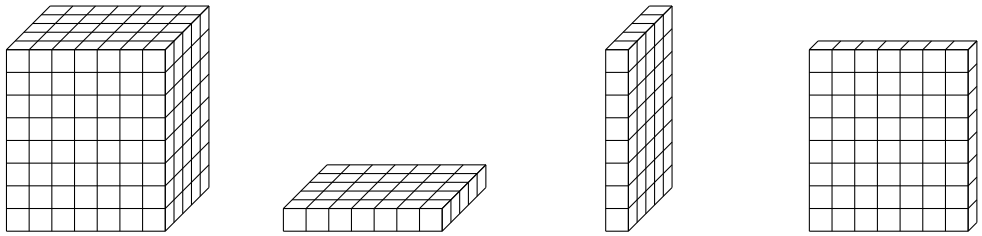


FIGURE 1.9: Tensor compression: (A) Three-way tensor, (B) Vertical compression, (C) Horizontal compression, (D) Frontal compression.

Considering a three-way tensor $4 \times 4 \times 2$, the frontal compression using *sum* aggregation function is equal to:

$$\mathbf{X}^1 = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 5 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{X}^2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 13 & 16 \\ 0 & 0 & 14 & 15 \end{bmatrix}, \mathbf{X}_{comp} = \begin{bmatrix} 3 & 3 & 1 & 0 \\ 6 & 4 & 0 & 0 \\ 0 & 0 & 13 & 16 \\ 0 & 1 & 14 & 15 \end{bmatrix}. \tag{1.7}$$

1.3.2 Tensor analysis and clustering approaches

Tensor data representation becomes a handy tool to represent data with complex structure. The three-way tensor data allow to preserve a natural composition of data and are used in different fields like recommender systems, medical fields, and social study [Henriques and Madeira, 2018]. There are various ways to analyze tensor data, in this work, we investigate the tensor-based approaches, and we will present different variants of tensor approaches task explored in the existing literature.

A large number of tensor-based methods have been proposed in the literature. Based on the survey classification proposed in [Henriques and Madeira, 2018, Kolda and Bader, 2009, De Lathauwer, 2009] and our investigations, we propose to classify the existing tensor-based methods to three groups considering the proprieties of approaches:(1) tensor factorization based approaches (2) stochastic approaches (3) low-rank approximation based approaches.There are some other tensor-based methods, which will be cited at the end of this section.

Tensor factorization based approaches

A large variety of tensor factorization methods was developed in the literature. In this section, we describe the most popular methods and offer a brief review of recent approaches.

CANDECOMP/PARAFAC Decomposition. CANDECOMP/PARAFAC (or CP) is one of the most popular tensor decomposition methods. It assumes that the N-way tensor can be expressed by the sum of a finite number of rank-one tensors [Hitchcock, 1927, 1928, Carroll and J., 1970, Harshman, 1970]. Like PCA, there is no algorithm to determine with guarantee the number of principal components (rank for tensor).

Let $\mathcal{X} \in \mathbb{R}^{n \times d \times v}$ be a three-way tensor. CP aims to decompose the tensor $\mathcal{X}$ to K components which represent the best approximation of $\mathcal{X}$ such as,

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\|, \text{ where } \hat{\mathcal{X}} = \sum_{k=1}^{K} \lambda_k \mathbf{a}_k \circ \mathbf{b}_k \circ \mathbf{c}_k.$$

The ALS (Alternating Least Squares) approach can be used to solve this optimization problem and find all components.

Tucker Decomposition. The Tucker decomposition, proposed by Tucker [1963], has become well-known by other names, namely Three-mode factor analysis (3MFA/Tucker3) [Tucker, 1966], Three-mode PCA (3MPCA) [Kroonenberg, 1983], N-mode PCA [Kapteyn et al., 1986], Higher-order SVD (HOSVD) [De Lathauwer et al., 2000], and N-mode SVD [Vasilescu and Terzopoulos, 2002]. It can be viewed as a form of higher-order PCA. It decomposes the tensor into a core tensor multiplied by a matrix along with each mode. For instance, considering a three-way tensor $\mathcal{X} \in \mathbb{R}^{n \times d \times v}$, the optimization problem can be written as follows :

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\|, \text{ where } \hat{\mathcal{X}} = \sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{q=1}^{Q} \mathcal{H}_{kpq} \mathbf{a}_k \circ \mathbf{b}_p \circ \mathbf{c}_q,$$

where $\mathcal{H} \in \mathbb{R}^{K \times P \times W}$ is a core tensor, $\mathbf{A} \in \mathbb{R}^{n \times K}$ related to the first mode, $\mathbf{B} \in \mathbb{R}^{d \times P}$ related to the second mode, and $\mathbf{C} \in \mathbb{R}^{v \times Q}$ associated with the third mode. Some algorithms based on the ALS approach has been developed to solve this optimization problem, such as TUCKALS2 and TUCKALS3 [Kapteyn et al., 1986, Kroonenberg, 1983].

Inspired by CP and Tucker decomposition, a lot of decomposition model was proposed. For instance, INDSCAL [Carroll and J., 1970] is a particular case of CP for three-way tensors that are symmetric in two modes. PARFAC2 [Harshman, 1972] is also a variant of CP which can be used with a set of matrices that have the same number of columns and different a number of rows. CANDELINC [Carroll et al., 1980] is a CP with linear constraints on one or more modes. DEDICOM [Harshman, 1994] considers multiple asymmetric relationships and decomposes the tensor with the objective of regrouping objects in clusters based on the discovered latent components. Harshman and Lundy [1996] proposed PARATUCK2, as its name suggests, it can be considered as a combination of CP and Tucker decomposition, and a generalization of DEDICOM where the row and column objects can be different sets. Finally, RSCAL [Nickel et al., 2011], can be considered as a relaxed version of DEDICOM.

Stochastic approaches.

Few works developed stochastic approaches for tensor clustering and co-clustering. In [Penga and Lib, 2011], the authors proposed an algorithm called ASI-T (Adaptive Subspace Iteration on Tensor) for multi-way (tensor) data clustering, and demonstrated that ASI-T is a special version of HOSVD. [Sra et al., 2008] proposed a Bergman tensor clustering and showed some proprieties and links with euclidian k-means.

In 2005 [Zhao and Zaki, 2005] proposed the first formulation of the tri-clustering for gene expression application. The data structure gene-sample-time is viewed as a multi-graphs. The proposed approach finds a set of bi-clusters, and then tri-clusters are defined by merging

similar bi-clusters. In [Schepers et al., 2006], the authors minimize the least-squares loss function between tensor data and a prediction of the bi-clustering model; for more details about stochastic approaches and tri-clustering algorithms please see [De Lathauwer, 2009, Ahmed et al., 2011, Guigoures et al., 2012, Tchagang et al., 2012, Mankad and Michailidis, 2014, Liu et al., 2015, Wu et al., 2018].

Low-rank approximation based approaches

Low-rank (LR) tensor approximation methods have become an important tool in multi-linear algebra problems, which are intractable comparing with classical approaches. LR based approaches are used on several applications; however, LR methods showed more effectiveness dealing with high-dimensional images (see [Grasedyck et al., 2013]). In [Yan et al., 2014], the authors propose an image-based process monitoring approach that is capable of handling both grayscale and color images. The proposed approach models the high-dimensional structure of the image data with tensors and employs low-rank tensor decomposition techniques to extract important monitoring features. In 2015, [Li et al., 2015] propose a Low-rank Tensor Decomposition based anomaly Detection (LTDD) algorithm for Hyperspectral images (HSI). LTDD is adapted to deal with sparse three-way tensors. [Zhang et al., 2015] developed Low-rank Tensor constrained Multiview Subspace Clustering (LT-MSC) approach. LT-MSC deals with multiple similarity matrices (views) structured as a three-way tensor. The proposed approach allows us to capture the global structure of all the views and explore the correlations within and across multiple views. More recently, in [Du et al., 2017], the authors proposed a novel HSI compression and reconstruction algorithm via PLTD, a low-rank tensor approximation algorithm. PLTD preserves the correlation among the spectral dimension, which allows a better reconstruction of images.

1.3.3 Tensor clustering applications

Recently, tensor-based analyses have been successfully performed in many areas. In this section, we present the most popular applications' fields and give a brief review of other less well-known applications [Kolda and Bader, 2009, Zhang et al., 2013].

Signal Processing

Signal data (times series) are used in different fields such as bio-medical, speech recognition, sensors data, etc. These data are not necessarily structured into a 2D matrix. In many cases, signals can depend on multiple entities; the *Electroencephalography*(EEG) signal is, for instance, generated with different *channels* and different time frequencies, leading to tensorial representation. Cong et al. [2015] presented a brief review of tensor decomposition methods applied to EEG data. In [Mahyari et al., 2017], the authors proposed an approach based on tensor representation for detecting dynamic network states from EEG.

Other works on medical fields proposed a tensor-based approach to model *Polyaffine* motion characterizing Pathological Left Ventricular Dynamics [McLeod et al., 2015]. In [Zhang et al., 2017] a tensor decomposition allowed to deal with heart sound classification. To go further, in signal processing context, the reader can refer to [Lim and Comon, 2010, Zhang et al., 2013, Sidiropoulos et al., 2017, Wang et al., 2017].

Images and Hyperspectral images

Recently, the hyperspectral images (HSI) has received a growing attention, due to their high quality and information. HSI can be represented by three-way data, unlike classical images which represented by 2d matrix where each entry is a pixel. In HSI, each image is composed by a multiple *bands* which are images generated according to electromagnetic spectrum. The compression and reconstruction of HSI have generated a lot of interest. Guo et al. [2013] proposed a rank-1 tensor decomposition for image noise reduction, using a top eigenvalue and reconstruction technique. Compared with the existing noise reduction methods such as the conventional channel-by-channel approaches, the proposed R1TD method improves the image reconstruction results in terms of both visual inspection and image quality indices. Du et al. [2017] developed a novel HSI compression and reconstruction algorithm via patch-based low-rank tensor approximation technique (PLTD), while Veganzones et al. [2016] designed a Nonnegative tensor CP decomposition for tensor data. There are also works about HSI restoration and anomaly detection based on tensor decomposition approaches [Wang et al., 2018, Zhang et al., 2016]. In the face recognition domain, several tensor-based approaches relying on tensor decomposition have been developed; see for instance [Cao et al., 2015, Hašan et al., 2008, Moberts et al., 2005, He et al., 2005, Zhang et al., 2016]. To go further, as a video is composed of a sequence of frames (or images), a tensor decomposition is also used for video recognition [Abdallah et al., 2007].

Recommender systems

Data derived from recommender systems can be easily structured as a tensor. In fact, these data generally link two objects *users* and *items*. Furthermore, other information are available about users, items, and also the interaction between users and items (rating, reviews) conducting to a tonsorial structure. Some works review the developed approaches using tensor data decomposition for recommender systems; e.g see for instance [Zhang et al., 2011, Ricci et al., 2012, Symeonidis, 2016].

The additional data available on recommender systems, well known as *Context-Aware Recommendation*, can be used to improve recommendation results. Thus, information about user age, sex, and occupation can allow us to improve results. On the other hand, information about an item can help recommendations. In fact, if we consider a movie recommendation system, information about movie genre and actors are useful [Wermser et al., 2011, Karatzoglou et al., 2010].

In some social networks, tags are an essential element. Posts, images, and videos tagging allow us to easily find information and images on social media such as Twitter and Instagram, respectively. In this context, a lot of works proposed tensor-based approaches for tag clustering [Rafailidis and Daras, 2013, Symeonidis, 2016]. Recently, tensor decomposition methods are used for tag completion, refinement, and correction in social media [Zhang et al., 2011, Tang et al., 2017, Tang et al., 2019].

Other applications

As pointed out in previous sections, the bio-medical area is conducive to use tensor-based approaches. Several works about epigenomics and microarray using tensor data representation

are proposed. For instance, in [Durham et al., 2018], the authors proposed a tensor decomposition method to treat epigenomics data imputation. The proposed PREDICTD algorithm provides reference imputed data and demonstrates the utility of tensor decomposition on the imputation of missing values. In [Feizi et al., 2017], they proposed a tensor bi-clustering approach based on spectral decomposition of the tensor. The objective of this work is finding one bi-cluster, the most important one based on eigenvalues. We can also cite the work of Hore et al. [2016], whose construct $individuals \times tissus \times genes$ tensor and used tensor decomposition for multi-tissue gene expression clustering.

Some works used tensor-based approaches to deal with semantic web data. In fact, in the context of RDF knowledge bases, data can be seen as a graph where nodes represent RDF resources, and edges correspond to RDF predicates that link resources. Thus, multiple graphs can be constructed, and tensor representation can be adopted for the semantic Web [Franz et al., 2009, Saha et al., 2008, Drumond et al., 2012].

1.4 Conclusion

We briefly reviewed some popular approaches leading to clustering and co-clustering methods. Concerning tensor data, which is the main focus of this thesis, we showed that there are many research dealing with tensor data. Most approaches, rely on tensor decomposition, stochastic methods, and low-rank approximation methods.

In terms of softwares, recently, many packages and libraries for tensor data and tensor decomposition-based methods were developed. **TensorLy**[1] [Kossaifi et al., 2019] a python package implementing popular tensor decomposition methods such as PARAFAC and Tucker Decomposition. **TensorD**[2] [Hao et al., 2018] is a tensor library in TensorFlow. It provides basic decomposition methods such as Tucker and CANDECOMP/PARAFAC (CP) decompositions, as well as new decomposition methods, for example, Pairwise Interaction Tensor Decomposition. In the sequel, we do not consider these approaches, however in our contributions (next chapters), and especially in our comparisons, we refer to these kinds of methods.

[1] http://tensorly.org/stable/index.html
[2] https://github.com/Large-Scale-Tensor-Decomposition/tensorD

Chapter 2

Latent Block Model for Tensor Data

2.1 Introduction

Co-clustering addresses the problem of simultaneous clustering of both dimensions of a data matrix. Many of the datasets encountered in data science are two-dimensional in nature and can be represented by a matrix. Classical clustering procedures seek to construct separately an optimal partition of rows (individuals) or, sometimes (features), of columns. In contrast, co-clustering methods cluster the rows and the columns simultaneously and organize the data into homogeneous blocks (after suitable permutations); see for instance [Dhillon et al., 2003, Govaert and Nadif, 2003, Govaert and Nadif, 2005, Govaert and Nadif, 2008, 2013, Salah and Nadif, 2017, 2018, Ailem et al., 2017, Labiod and Nadif, 2011]. Methods of this kind have practical importance in a wide variety of applications where data are typically organized in two-way tables. However, in modern datasets, instead of collecting data on every individual-feature pair, we may collect supplementary individual or item information leading to tensor representation. This kind of data has emerged in many fields such as recommender systems where the data are collected on multiple items rated by multiple users, information about users and items is also available yielding as a tensor rather than a data matrix.

Despite the great interest for co-clustering techniques on the one hand and the tensor representation on the other, few works tackles co-clustering from tensor data. We mention the work based on Minimum Bregman information (MBI) to carry out co-clustering [Banerjee et al., 2005] and the *General Tensor Spectral Co-clustering* (GTSC) method suitable to non-negative tensor data [Wu et al., 2016]. Other approaches can be cited although the goal is not exactly co-clustering but only extracting a bicluster. For instance, in [Feizi et al., 2017] the authors aim to extract a bicluster composed of 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 the row 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 -row and column- are different and with model-based co-clustering. To this end, we rely on the latent block model [Govaert and Nadif, 2013] for its flexibility to consider any type of data matrices.

In this chapter, we propose a co-clustering model for tensor data, where clustering of row (indexed from $i = 1$ to n) and column (indexed from $j = 1$ to d) entities is done not only on principal relation matrix but on tensor including multiple covariates and/or relations between entities. The proposed model can also be viewed as multi-way clustering approach where each slice (indexed from $b = 1$ to v) of the third dimension of the tensor represents a relation or covariate (see Figure 2.1).Thereby the purpose to simultaneously discover the

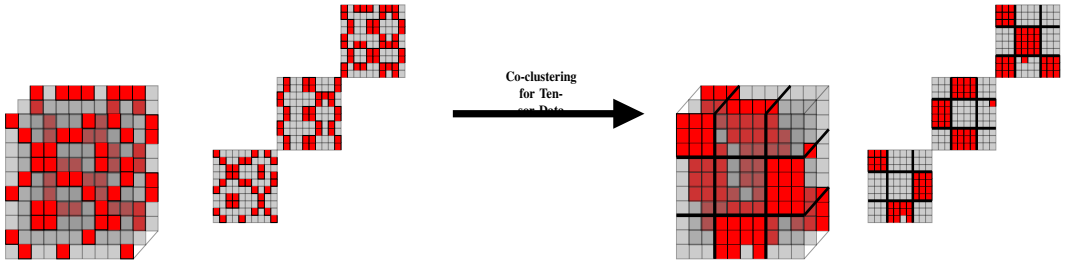


FIGURE 2.1: Goal of co-clustering for Binary Tensor data.

row (indexed from $k = 1$ to g) and column (indexed from $\ell = 1$ to m) clusters and the relationship between these clusters for all slices. To achieve this, we propose to extend Latent block model (LBM) to tensor data referred to as TLBM. This model is suitable for several applications. Our first investigation has appeared recently as a paper published in [Boutalbi et al., 2019a]. In the present manuscript, we delve in-depth into this idea and present several new theoretical and empirical results. The main contributions of this chapter are summarized as follows : (i) we propose an extension of latent block model for tensor data (TLBM) (ii) we show its flexibility to be applied with different types of data (iii) we derive a variational EM and a hard version for co-clustering.

The remainder of this chapter is organized as follows. Section 2.2 describes classical latent block model and presents its extension TLBM. Section 2.3 details the proposed algorithm variational EM for co-clustering of tensor data. In section 2.4, we present a hard version of the proposed algorithm and evaluate its performances. Section 2.5 presents experimental results on the synthetic and real word datasets. Section 2.6 concludes this chapter and provides some directions for future work.

2.2 Extension of Latent block model for tensors

In this section, we introduce the Latent Block Model (LBM), and we detail the variational EM and the Classification EM algorithms used for parameters estimation for LBM. Then we present our contribution to Tensor LBM, an extension of the LBM model to tensor data.

2.2.1 Latent block model

As introduced in section 1.2.4, the latent block model [Govaert and Nadif, 2013], given a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, assumes that there is a couple of partitions $(\mathbf{z}, \mathbf{w})$ where $\mathbf{z}$ is partitioned in g clusters on the set of rows I and $\mathbf{w}$ is partitioned in m clusters on the set of columns J, such that each element x_{ij} belonging to the block $k\ell$ is generated according to a probability distribution, where k represents the class of row i, while ℓ represents the class of column j. This model is based on the following assumptions:

- The univariate random variables x_{ij} are considered independent given the row partition $\mathbf{Z}$ and column partition $\mathbf{W}$.
- The latent variables $z_1, ..., z_n, w_1, ..., w_d$ are assumed to be independent $p(\mathbf{Z}, \mathbf{W}) = p(\mathbf{Z})p(\mathbf{W})$ where $p(\mathbf{Z}) = \prod_i^n p(z_i) = \prod_{i,k} \pi_k^{z_{ik}}$ and $p(\mathbf{W}) = \prod_j^d p(w_j) = \prod_{j,\ell} \rho_\ell^{w_{j\ell}}$.

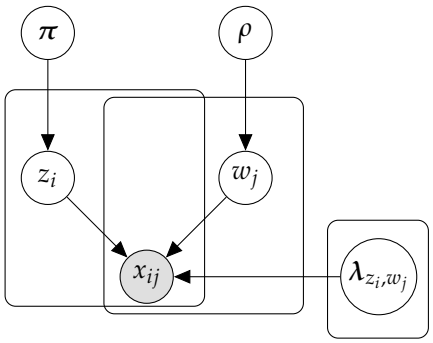


FIGURE 2.2: LBM graphical model.

- For all i, the distribution $p(z_i)$ is the multinomial distribution $\mathcal{M}(\pi_1, ..., \pi_g)$ and does not depend on i. Similarly, for all j, the distribution of $p(w_j)$ is the multinomial distribution $\mathcal{M}(\rho_1, ..., \rho_m)$ and does not depend on j.

The probability density function (*pdf*) of the latent block model can be written as follows:

$$f(\mathbf{X}, \mathbf{Z}, \mathbf{W}, \mathbf{\Omega}) = \sum_{(\mathbf{z},\mathbf{w})\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,\ell} \rho_\ell^{w_{j\ell}} \prod_{i,j,k,\ell} \left(\Phi(x_{ij}; \lambda_{k\ell})\right)^{z_{ik}w_{j\ell}}, \tag{2.1}$$

where $\mathbf{\Omega} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\lambda})$ are the unknown parameters of LBM 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 proportions of clusters and $\boldsymbol{\lambda} = (\lambda_{k\ell}; k = 1, ..., g; \ell = 1, ..., m)$ where $\lambda_{k\ell}$ represents the parameters of the distribution Φ. The classification log-Likelihood takes the following form:

$$L_C(\mathbf{Z}, \mathbf{W}, \mathbf{\Omega}) = \sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell + \sum_{i,j,k,\ell} z_{ik} w_{j\ell} \log(\Phi(x_{ij}; \lambda_{k\ell})). \tag{2.2}$$

The graphical model is depicted in figure 2.2 and the generative process of data according LBM is described in algorithm 1.

Algorithm 1: Generative process of `LBM` model

Input: n, d, g, m, $\boldsymbol{\pi}$, $\boldsymbol{\rho}$, $\boldsymbol{\lambda}$

for $i \leftarrow 1$ **to** n **do**
  Generate the row label z_i according to $\mathcal{M}(\pi_1, \ldots, \pi_g)$

for $j \leftarrow 1$ **to** d **do**
  Generate the column label w_j according to $\mathcal{M}(\rho_1, \ldots, \rho_m)$

for $i \leftarrow 1$ **to** n **and** $j \leftarrow 1$ **to** d **do**
  Generate a entry x_{ij} according to the density $\Phi(x_{ij}; \boldsymbol{\lambda}_{z_i,w_j})$.

return Data matrix $\mathbf{X}$, $\mathbf{z}$ and $\mathbf{w}$

Assuming that the complete data are composed by $(\mathbf{X}, \mathbf{Z}, \mathbf{W})$, the complete data log-likelihood function can be written as follows :

$$\begin{aligned} L_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega}) &= \log f(\mathbf{X}, \mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega}) \\ &= \log \prod_{i,k} \pi_k^{z_{ik}} + \log \prod_{j,\ell} \rho_\ell^{w_{j\ell}} + \log \prod_{i,j,k,\ell} \Phi(x_{ij}; \boldsymbol{\lambda}_{k\ell}) \\ &= \sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell + \sum_{i,j,k,\ell} z_{ik} w_{j\ell} \log(\Phi(x_{ij}; \boldsymbol{\lambda}_{k\ell})). \end{aligned} \tag{2.3}$$

The log-likelihood can be decomposed into three terms. The two first terms depend on row and column clusters proportion respectively. The third one depends on the pdf of each co-cluster.

To estimate the parameters $\boldsymbol{\Omega}$ using MLE, we can use the expectation-maximization (EM) algorithm. The E-step consists of computing the posteriori probabilities of the missing labels $\mathbf{z}$ and $\mathbf{w}$. The M-step is to updating the parameters by maximizing the expectation of the complete data log-likelihood $L_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega})$, defined as follows:

$$\mathbb{E}(L_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega}) | \boldsymbol{\Omega}^{(t)}, \mathbf{X}) = \sum_{ik} \tilde{z}_{ik}^{(t)} \log \pi_k + \sum_{j\ell} \tilde{w}_{j\ell}^{(t)} \log \rho_\ell + \sum_{i,j,k,\ell} \tilde{e}_{i,j,k,\ell}^{(t)} \log \Phi(x_{ij}; \boldsymbol{\lambda}_{k\ell}), \tag{2.4}$$

where $\tilde{z}_{ik}^{(t)} = \mathbb{E}(z_{ik} | \mathbf{x}_i, \boldsymbol{\Omega}^{(t)}) = p(z_{ik} | \mathbf{x}_i, \boldsymbol{\Omega}^{(t)})$, $\tilde{w}_{j\ell}^{(t)} = \mathbb{E}(w_{j\ell} | \mathbf{x}_j, \boldsymbol{\Omega}^{(t)}) = p(z_{j\ell} | \mathbf{x}_j, \boldsymbol{\Omega}^{(t)})$, and $\tilde{e}_{ikj\ell}^{(t)} = \mathbb{E}(z_{ik} w_{j\ell} | x_{ij}, \boldsymbol{\Omega}^{(t)}) = p(z_{ik} w_{j\ell} | x_{ij}, \boldsymbol{\Omega}^{(t)})$. Unfortunately, the double unknown data variable $\mathbf{Z}$ and $\mathbf{W}$ in e makes the maximization of $\mathbb{E}(L_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega}) | \boldsymbol{\Omega}^{(t)}, \mathbf{X})$ more difficult than of the classical mixture model.

To solve this problem, a mean-field variational EM (VEM) algorithm can be used for inferences. The objective is to appoximate the true posterior probability $p(\mathbf{Z}, \mathbf{W} | \mathbf{X}, \boldsymbol{\Omega}^{(t)})$ with a more tractable distribution $q(\mathbf{Z}, \mathbf{W}) = p(\mathbf{Z}) p(\mathbf{W})$. Then, using the [Neal and Hinton, 1998] interpretation of the EM algorithm, the mean-field VEM algorithm is equivalent to mximize with respect to q and $\boldsymbol{\Omega}$ the following soft co-clustering criteria:

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

where, $H(\tilde{Z}) = -\sum_{ik} \tilde{z}_{ik} \log \tilde{z}_{ik}$ and $H(\tilde{W}) = -\sum_{j\ell} \tilde{w}_{j\ell} \log \tilde{w}_{j\ell}$ are respectively the entropy of the unknown variables $\mathbf{Z}$ and $\mathbf{W}$ where $\tilde{z}_{ik} = q(z_{ik} = 1)$ and $\tilde{w}_{j\ell} = q(w_{j\ell} = 1)$, $L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega})$ is the fuzzy complete-data log-likelihood. The maximization of the function $F_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega})$ can be obtained by alternating two steps: (i) for given partition of variables, we optimize the partition of objects and the model's parameters until convergence; (ii) for a given partition of objects, we update the variable partition and the model's parameters until convergence. These two steps are repeated until convergence. This algorithm is described below (see Algorithm 2).

Algorithm 2: LBVEM

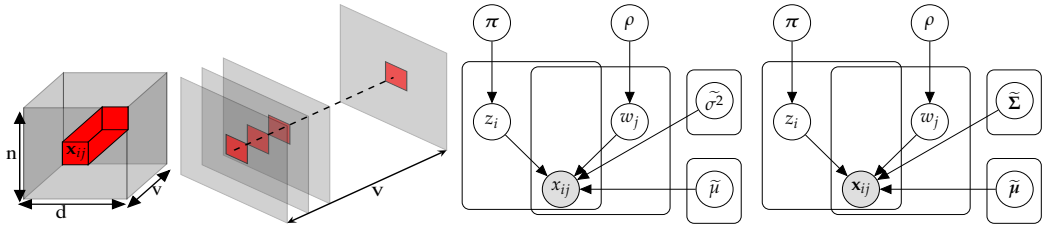
Input: $\mathbf{X}$, g, m.
Initialization $(\mathbf{Z}, \mathbf{W})$ randomly, compute $\mathbf{\Omega}$
repeat
  repeat
    - **E-step : Compute** $\tilde{z}_{ik}$
    - **M-step : Compute the parameters of the model** $\mathbf{\Omega}$

  until *convergence*;
  repeat
    - **E-step : Compute** $\tilde{w}_{j\ell}$
    - **M-step : Compute the parameters of the model** $\mathbf{\Omega}$

  until *convergence*;
until *convergence*;
return Z, W, Ω

2.2.2 Latent Block Model for Tensor data (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 [Feizi et al., 2017, Wu et al., 2016]. Unlike classical LBM which considers data matrix $\mathbf{X} = [x_{ij}] \in \mathbb{R}^{n \times d}$, TLBM considers 3D data matrix $\mathcal{X} = [\mathbf{x}_{ij}] \in \mathbb{R}^{n \times d \times v}$ where n is the number of rows, d the number of columns, and v the number of covariates. Figure 2.3a presents the data structure. Note that in our cases, a co-cluster is a parallelepiped.



(A) (B) (C)

FIGURE 2.3: (a) Data structure, (b) Gaussian LBM with $\widetilde{\mu} = \{\mu_{11}, \ldots, \mu_{gm}\}, \widetilde{\sigma^2} = \{\sigma^2_{11}, \ldots, \sigma^2_{gm}\}$ where $\forall k, \ell, \quad \mu_{k\ell}, \sigma^2_{k,\ell} \in \mathbb{R}$, (c) Gaussian TLBM with $\widetilde{\boldsymbol{\mu}} = \{\boldsymbol{\mu}_{11}, \ldots, \boldsymbol{\mu}_{gm}\}, \widetilde{\boldsymbol{\Sigma}} = \{\boldsymbol{\Sigma}_{11}, \ldots, \boldsymbol{\Sigma}_{gm}\}$ where $\forall k, \ell, \quad \boldsymbol{\mu}_{k\ell} \in \mathbb{R}^{v \times 1}$ and $\boldsymbol{\Sigma}_{k\ell} \in \mathbb{R}^{v \times v}$.

Continuous data. In this case, we can assume $\Phi(\mathbf{x}_{ij}; \boldsymbol{\lambda}_{k\ell})$ as a multivariate normal distribution with mean vector $\boldsymbol{\mu}_{k\ell}^\top = (\mu^1_{k\ell}, \ldots, \mu^v_{k\ell})$ and covariance matrix $\boldsymbol{\Sigma}_{k\ell}$ of size $v \times v$. The parameter $\mathbf{\Omega}$ is formed by π, $\boldsymbol{\rho}$ and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_{11}, \ldots, \boldsymbol{\lambda}_{gm})$. Hence, $\Phi(\mathbf{x}_{ij}; \boldsymbol{\lambda}_{k\ell})$ takes the following form.

$$\frac{1}{(2\pi)^{n/2}|\Sigma_{k\ell}|^{0.5}} \exp\left\{-\frac{1}{2}(\mathbf{x}_{ij} - \boldsymbol{\mu}_{k\ell})^\top \boldsymbol{\Sigma}_{k\ell}^{-1}(\mathbf{x}_{ij} - \boldsymbol{\mu}_{k\ell})\right\}$$

and,

$$L_C(\mathbf{Z},\mathbf{W},\boldsymbol{\Omega}) = \sum_{i,k} z_{ik}\log\pi_k + \sum_{j,\ell} w_{j\ell}\log\rho_\ell - \frac{1}{2}\sum_{k,\ell} z_{\cdot k}w_{\cdot \ell}\log|\Sigma_{k\ell}| - \frac{1}{2}\sum_{i,j,k,\ell} z_{ik}w_{j\ell}(\mathbf{x}_{ij}-\boldsymbol{\mu}_{k\ell})^\top \boldsymbol{\Sigma}_{k\ell}^{-1}(\mathbf{x}_{ij}-\boldsymbol{\mu}_{k\ell}). \tag{2.5}$$

The graphical models of Gaussian LBM and Gaussian TLBM are depicted respectively in figures 2.3b and 2.3c. With Gaussian LBM, for each block (k,ℓ), $x_{ij} \in \mathbb{R} \sim \mathcal{G}(\mu_{k\ell},\sigma^2_{k\ell})$ while with Gaussian TLBM, $\mathbf{x}_{ij} \in \mathbb{R}^{v\times 1} \sim \mathcal{G}(\boldsymbol{\mu}_{k\ell},\boldsymbol{\Sigma}_{k\ell})$ allowing to take into account the covariances between all v variables.

Binary data. In this case, we can consider an extension of the Bernoulli LBM (Bernoulli TLBM), thereby $\boldsymbol{\mu}_{k\ell}$ is a probability vector. Specifically, assuming the concept of conditional independence (independence per block) which is the basis for many statistical models Φ is given by

$$\Phi(\mathbf{x}_{ij};\boldsymbol{\lambda}_{k\ell}) = \prod_{b=1}^{v}(\mu^b_{k\ell})^{x^b_{ij}}(1-\mu^b_{k\ell})^{1-x^b_{ij}},$$

and the classification log-likelihood can be written as

$$L_C(\mathbf{Z},\mathbf{W},\boldsymbol{\Omega}) = \sum_{i,k} z_{ik}\log\pi_k + \sum_{j,\ell} w_{j\ell}\log\rho_\ell + \sum_{k\ell} z_{\cdot k}w_{\cdot\ell}\sum_{b=1}^{v}\log(1-\mu^b_{k\ell}) + \sum_{i,j,k,\ell} z_{ik}w_{j\ell}\left(\sum_{b=1}^{v} x^b_{ij}\log\frac{\mu^b_{k\ell}}{1-\mu^b_{k\ell}}\right) \tag{2.6}$$

with $z_{\cdot k} = \sum_i z_{ik}$ and $w_{\cdot\ell} = \sum_j w_{j\ell}$.

Count data (also known as a cross tabulation). In this case, we can consider an extension of the Poisson LBM (Poison TLBM), thereby $\boldsymbol{\lambda}^b_{ij}$ is a vector of parameters. Like with Bernoulli TLBM we assume the conditional independence, thereby Φ is given by

$$\Phi(\mathbf{x}_{ij};\boldsymbol{\lambda}_{k\ell}) = \prod_{b=1}^{v}\frac{e^{-\lambda^b_{ij}}{\lambda^b_{ij}}^{x^b_{ij}}}{x^b_{ij}!},$$

where $\lambda^b_{ij} = x^b_{i\cdot}x^b_{\cdot j}\sum_{k,\ell} z_{ik}w_{j\ell}\gamma^b_{k\ell}$ with the margins $x^b_{i\cdot} = \sum_j x^b_{ij}$ and $x^b_{\cdot j} = \sum_i x^b_{ij}$ and the block effects $\gamma_{k\ell}$. Therefore the parameter $\boldsymbol{\Omega}$ to be estimated is formed by $\boldsymbol{\pi}$, $\boldsymbol{\rho}$ and $\gamma = (\gamma_{11},\ldots,\gamma_{gm})$ where $\gamma_{k\ell} = (\gamma^1_{k\ell},\ldots,\gamma^v_{k\ell})$. The generative process is described in algorithm 3; TLBM is flexible and can be used with different types of data.

The classification log-likelihood (up to a constant) can be written as

$$L_C(\mathbf{Z},\mathbf{W},\boldsymbol{\Omega}) = \sum_{i,k} z_{ik}\log(\pi_k) + \sum_{j,\ell} w_{j\ell}\log\rho_\ell + \sum_{i,j,k,\ell} z_{ik}w_{j\ell}\sum_b\left(-x^b_{i\cdot}x^b_{\cdot j}\gamma^b_{k\ell} + x^b_{ij}\log(\gamma^b_{k\ell})\right). \tag{2.7}$$

Algorithm 3: Generative process of `Tensor LBM` model

Input: $n, d, g, m, \boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\lambda}$

for $i \leftarrow 1$ **to** n **do**
  Generate the row label z_i according to $\mathcal{M}(\pi_1, \ldots, \pi_g)$

for $j \leftarrow 1$ **to** d **do**
  Generate the column label w_j according to $\mathcal{M}(\rho_1, \ldots, \rho_m)$

for $i \leftarrow 1$ **to** n **and** $j \leftarrow 1$ **to** d **do**
  Generate a vector $\mathbf{x}_{ij}$ according to the density $\Phi(\mathbf{x}_{ij}; \lambda_{k\ell})$.

return Tensor $\boldsymbol{\mathcal{X}}$, $\mathbf{z}$ and $\mathbf{w}$

In table 2.1, we repport the expressions of $L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega})$ according to various distributions.

Next, we propose a generic co-clustering algorithm able to propose solutions for different types of tensors data encountered in practice.

2.3 Variational EM algorithm for TLBM

To estimate $\boldsymbol{\Omega}$, the EM algorithm [Dempster et al., 1977] is a candidate for this task. It maximizes the log-likelihood $f(\boldsymbol{\mathcal{X}}, \boldsymbol{\Omega})$ 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 $\boldsymbol{\mathcal{X}}$. Unfortunately, difficulties arise owing to the dependence structure among the variables $\mathbf{x}_{ij}$ of the model. To solve this problem an approximation using the interpretation of the EM algorithm can be proposed; see, e.g., [Govaert and Nadif, 2005, Govaert and Nadif, 2008, 2013]. More precisely, the authors rely on the variational approach which consists of approximating the true likelihood by another expression using the following independence assumption:

$$P(z_{ik} = 1, w_{j\ell} = 1|\boldsymbol{\mathcal{X}}) = \underbrace{P(z_{ik} = 1|\boldsymbol{\mathcal{X}})}_{\tilde{z}_{ik}} \underbrace{P(w_{j\ell} = 1|\boldsymbol{\mathcal{X}})}_{\tilde{w}_{j\ell}}.$$

Hence, the aim is to maximize the following lower bound of the log-likelihood criterion:

$$F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}; \boldsymbol{\Omega}) = L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega}) + H(\tilde{\mathbf{Z}}) + H(\tilde{\mathbf{W}}) \tag{2.8}$$

where $\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}$ are fuzzy matrices and $L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}; \boldsymbol{\Omega})$ is the fuzzy complete data log-likelihood and

$$\begin{cases} H(\tilde{\mathbf{Z}}) = -\sum_{i,k} \tilde{z}_{ik} \log \tilde{z}_{ik} \\ H(\tilde{\mathbf{W}}) = -\sum_{j,\ell} \tilde{w}_{j\ell} \log \tilde{w}_{j\ell}. \end{cases}$$

TABLE 2.1: Expression of $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega})$ according various TLBM.

TLBM	$F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega})$
Gaussian	$\sum_{i,k} \tilde{z}_{ik} \log(\pi_k) + \sum_{j,\ell} \tilde{W}_{j\ell} \log \rho_\ell + H(\tilde{\mathbf{Z}}) + H(\tilde{\mathbf{W}})$ $-\frac{1}{2} \sum_{k,\ell} \tilde{z}_{.k} \tilde{w}_{.\ell} \log \lvert\Sigma_{k\ell}\rvert - \frac{1}{2} \sum_{i,j,k,\ell} \tilde{z}_{ik} \tilde{w}_{j\ell} (\mathbf{x}_{ij} - \boldsymbol{\mu}_{k\ell})^\top \boldsymbol{\Sigma}_{k\ell}^{-1} (\mathbf{x}_{ij} - \boldsymbol{\mu}_{k\ell})$
Bernoulli	$\sum_{i,k} \tilde{z}_{ik} \log(\pi_k) + \sum_{j,\ell} \tilde{w}_{j\ell} \log \rho_\ell + H(\tilde{\mathbf{Z}}) + H(\tilde{\mathbf{W}})$ $+ \sum_{k,\ell} \tilde{z}_{.k} \tilde{w}_{.\ell} \sum_b \log(1 - \mu_{k\ell}^b) + \sum_{i,j,k,\ell} \tilde{z}_{ik} \tilde{w}_{j\ell} \left(\sum_b x_{ij}^b \log \frac{\mu_{k\ell}^b}{1-\mu_{k\ell}^b} \right)$
Poisson	$\sum_{i,k} \tilde{z}_{ik} \log(\pi_k) + \sum_{j,\ell} \tilde{w}_{j\ell} \log \rho_\ell + H(\tilde{\mathbf{Z}}) + H(\tilde{\mathbf{W}})$ $+ \sum_{i,j,k,\ell} \tilde{z}_{ik} \tilde{w}_{j\ell} \sum_b \left(-x_{i.}^b x_{.j}^b \gamma_{k\ell}^b + x_{ij}^b \log(\gamma_{k\ell}^b) \right)$

The maximization of $F_C(\tilde{\mathbf{z}}, \tilde{\mathbf{w}}, \boldsymbol{\Omega})$ can be reached by realizing the three successive optimizations:

$$\begin{cases} \arg\max_{\tilde{\mathbf{Z}}} F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega}), \\ \arg\max_{\tilde{\mathbf{W}}} F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega}), \\ \arg\max_{\boldsymbol{\Omega}} F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega}). \end{cases}$$

In what follows, we detail the Expectation (E) and Maximization (M) step of the Variational EM algorithm for tensor data. We can propose a generic version of Tensor co-clustering considering an independence between slices. Thus, the fuzzy log-likelihood takes the following form :

$$L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega}) = \sum_{i,k} \tilde{z}_{ik} \log \pi_k + \sum_{j,\ell} \tilde{w}_{j\ell} \log \rho_\ell + \sum_{i,j,k,\ell} \tilde{z}_{ik} \tilde{w}_{j\ell} \sum_b \log(\Phi(x_{ij}^b; \boldsymbol{\lambda}_{k\ell}^b)). \quad (2.9)$$

2.3.1 E-step

The E-step consists of computing, for all i, k, j, ℓ the posterior probabilities $\tilde{z}_{ik}$ and $\tilde{w}_{j\ell}$ maximizing $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega})$ given the estimated parameters $\boldsymbol{\Omega}_{k\ell}$. It is easy to show that, the posterior probability $\tilde{z}_{ik}$ maximizing $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega})$ (See Appendix A) is given by:

$$\tilde{z}_{ik} \propto \pi_k \exp\left(\sum_{j,\ell} \tilde{w}_{j\ell} \log\left(\Phi(\mathbf{x}_{ij}; \boldsymbol{\lambda}_{k\ell}) \right) \right).$$

In the same manner, the posterior probability $\tilde{w}_{j\ell}$ is given by:

$$\tilde{w}_{j\ell} \propto \rho_\ell \exp\left(\sum_{i,k} \tilde{z}_{ik} \log\left(\Phi(\mathbf{x}_{ij}; \boldsymbol{\lambda}_{k\ell})\right)\right).$$

2.3.2 M-step

Given the previously computed posterior probabilities $\tilde{\mathbf{Z}}$ and $\tilde{\mathbf{W}}$, the M-step consists of updating , $\forall k, \ell$, the parameters π_k, ρ_ℓ, $\boldsymbol{\mu}_{k\ell}$ and $\boldsymbol{\lambda}_{k\ell}$ maximizing $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega})$. The estimated parameters are defined as follows. First, taking into account the constraints $\sum_k z_{ik} = 1$ and $\sum_\ell w_{j\ell} = 1$, it is easy to show that $\hat{\pi}_k = \frac{\sum_i \tilde{z}_{ik}}{n} = \frac{\tilde{z}_{.k}}{n}$ and $\hat{\rho}_\ell = \frac{\sum_j \tilde{w}_{j\ell}}{d} = \frac{\tilde{w}_{.\ell}}{d}$. Secondly, the update of $\boldsymbol{\lambda}_{k\ell}$ depends on the choice of Φ (See Appendix B).

Gaussian TLBM. With this model, $\boldsymbol{\lambda}_{k\ell}$ is formed by $(\boldsymbol{\mu}_{k\ell}, \boldsymbol{\Sigma}_{k\ell})$ where $\boldsymbol{\mu}_{k\ell}$ is the mean vector and it is easy to show that the estimation of mean vector $\hat{\boldsymbol{\mu}}_{k\ell}$ is given by $\frac{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}\mathbf{x}_{ij}}{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}}$, and thereby deduce,

$$\hat{\boldsymbol{\Sigma}}_{k\ell} = \frac{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}(\mathbf{x}_{ij} - \hat{\boldsymbol{\mu}}_{k\ell})(\mathbf{x}_{ij} - \hat{\boldsymbol{\mu}}_{k\ell})^\top}{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}}.$$

Bernoulli TLBM. It easy to show that the update of $\boldsymbol{\lambda}_{k\ell}$ can be performed by the update of $\lambda^b_{k\ell}$'s separately. Thereby, from (2.6). For each triplet (k, ℓ, b), the partial derivative of

$$z_{.k} w_{.\ell} \log(1 - \mu^b_{k\ell}) + \sum_{i,j} z_{ik} w_{j\ell} \left(x^b_{ij} \log \frac{\mu^b_{k\ell}}{1 - \mu^b_{k\ell}}\right),$$

set to 0 leads to $\hat{\mu}^b_{k\ell} = \frac{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}x^b_{ij}}{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}}$. Hence $\boldsymbol{\lambda}_{k\ell}$ which is a probability vector is given by $\frac{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}\mathbf{x}_{ij}}{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}}$.

Poisson TLBM. Similarly, we can update $\gamma^b_{k\ell}$ for $\gamma_{k\ell}$. we have $\forall k, \ell, b$,

$$\begin{aligned}\hat{\gamma}^b_{k\ell} &= \arg\max_{\gamma_{k\ell}} \sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}(x^b_{ij} \log \gamma_{k\ell} - x^b_{i.}x^b_{.j}\gamma_{k\ell}) \\ &= \frac{\sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}x^b_{ij}}{\sum_i z_{ik}x^b_{i.} \sum_j w_{j\ell}x^b_{.j}} = \frac{x^b_{k\ell}}{x^b_{k.}x^b_{.\ell}},\end{aligned}$$

where,

$$x^b_{k\ell} = \sum_{i,j} \tilde{z}_{ik}\tilde{w}_{j\ell}x^b_{ij}, x^b_{k.} = \sum_i \tilde{z}_{ik}x^b_{i.}, x^b_{.\ell} = \sum_j \tilde{w}_{j\ell}x^b_{.j}.$$

The proposed algorithm for tensor data, referred to as `VEM-T` in Algorithm 4, alternates the two previously described steps Expectation-Maximization. At the convergence, a hard co-clustering is deduced from the posterior probabilities.

Algorithm 4: VEM−T

Input: $\mathcal{X}$, g, m.

Initialization $(\mathbf{Z}, \mathbf{W})$ randomly, compute $\boldsymbol{\Omega}$

repeat

E-Step

- **Compute** $\tilde{z}_{ik}$ **using**
  $\tilde{z}_{ik} \propto \pi_k \exp\left(\sum_{j,\ell} \tilde{w}_{j\ell} \log\left(\Phi(\mathbf{x}_{ij}; \boldsymbol{\lambda}_{k\ell})\right)\right)$
- **Compute** $\tilde{w}_{j\ell}$ **using**
  $\tilde{w}_{j\ell} \propto \rho_\ell \exp\left(\sum_{i,k} \tilde{z}_{ik} \log\left(\Phi(\mathbf{x}_{ij}; \boldsymbol{\lambda}_{k\ell})\right)\right)$

M-Step

Update $\boldsymbol{\Omega}$

until *convergence*;

return Z, W, Ω

2.4 Classification Maximum Likelihood approach

Another Likelihood-based approach to clustering besides the mixture likelihood is what is sometimes called the *Classification Maximum Likelihood* (CML) approach [Celeux and Govaert, 1992, Govaert and Nadif, 1996]. Unlike the *Maximum Likelihood* (ML) approach which aims to maximize log-likelihood, with the CML approach, $(\mathbf{Z}, \mathbf{W}, \Omega)$ are chosen to maximize the complete data log-likelihood $L_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega})$ (1.2). Doing so, the maximization can be obtained by alternating the three following computations:

$$\begin{cases} \arg\max_{\mathbf{Z}} F_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega}), \\ \arg\max_{\mathbf{W}} F_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega}), \\ \arg\max_{\boldsymbol{\Omega}} F_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega}). \end{cases}$$

These optimizations can be performed by using the Classification EM algorithm proposed in [Govaert and Nadif, 2008]. It is a direct clustering algorithm which consists of inserting a classification step (C-step) between E-step and M-step. The principal steps of the algorithm, which we refer as CEM−T, are reported in Algorithm 5. Note that in M-step all the update formulas can be used by replacing $\tilde{z}_{ik}$ by $z_{ik} \in \{0,1\}$ and $\tilde{w}_{j\ell}$ by $w_{j\ell} \in \{0,1\}$. In other words, the update is done by co-cluster.

Note with the CML approach, we can establish some connections with popular algorithms. Next, we show the connection in the case of contingency tables.

CEM−T for count data

When we consider Poisson TLBM, we have seen that the computation of $\hat{\boldsymbol{\gamma}}^b = \{\hat{\gamma}_{k\ell} | k = 1, \ldots, g; \ell = 1, \ldots, m; b = 1, \ldots, v\}$ maximizing L_C leads to $\hat{\gamma}^b_{k\ell} = \frac{x^b_{k\ell}}{x^b_{k\cdot} x^b_{\cdot\ell}}$ for all b, k, ℓ.

Algorithm 5: CEM−T

Input: $\mathcal{X}$, g, m.
Initialization $(\mathbf{Z}, \mathbf{W})$ randomly, compute $\mathbf{\Omega}$
repeat
E-Step:
- **Compute** $\tilde{z}_{ik}$ **using**
  $\tilde{z}_{ik} \propto \pi_k \exp\left(\sum_{j,\ell} \tilde{w}_{j\ell} \log\left(\Phi(\mathbf{x}_{ij}; \boldsymbol{\lambda}_{k\ell})\right)\right)$
- **Compute** $\tilde{w}_{j\ell}$ **using**
  $\tilde{w}_{j\ell} \propto \rho_\ell \exp\left(\sum_{i,k} \tilde{z}_{ik} \log\left(\Phi(\mathbf{x}_{ij}; \boldsymbol{\lambda}_{k\ell})\right)\right)$

C-Step:
- **Compute** $z_{ik} = \arg\max_{k'} \tilde{z}_{ik'} \quad \forall k' = 1, \ldots, g$
- **Compute** $w_{j\ell} = \arg\max_{\ell'} \tilde{w}_{j\ell'} \quad \forall \ell' = 1, \ldots, m$

M-Step:
Update $\mathbf{\Omega}$
until *convergence*;
return Z, W, Ω

Then plugging $\hat{\gamma}^b_{k\ell}$ in (2.7), the complete data log-likelihood $L_C(\mathbf{Z}, \mathbf{W}, \hat{\boldsymbol{\gamma}})$ becomes

$$L_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\gamma}) = \sum_{b=1}^{v} \sum_{k=1}^{g} \sum_{\ell=1}^{m} x^b_{k\ell} \log \frac{x^b_{k\ell}}{x^b_{k.} x^b_{.\ell}} - \sum_{b=1}^{v} N^b, \tag{2.10}$$

where $N^a = \sum_{i,j} x^b_{ij}$. The distribution that can be associated to $\mathbf{z}$ and $\mathbf{w}$ is the distribution defined by $p^b_{k\ell} = \frac{x^b_{k\ell}}{N^b}$ for all b, k, ℓ. The row and column margins are respectively defined by $p^b_{k.} = \frac{x^a_{k.}}{N^b}$ and $p^b_{.\ell} = \frac{x^b_{.\ell}}{N^b}$. Plugging these expressions in (2.10), the complete data log-likelihood can be expressed as follows:

$$\sum_{b=1}^{v} N^b \sum_{k=1}^{g} \sum_{\ell=1}^{m} p^b_{k\ell} \log \frac{p^b_{k\ell}}{p^b_{k.} p^b_{.\ell}} - \sum_{b=1}^{v} N^b (1 + \log N^b),$$

and its maximization is equivalent to the maximization of the total mutual information $\sum_{b=1}^{v} \sum_{k,\ell} p^b_{k\ell} \log \frac{p^b_{k\ell}}{p^b_{k.} p^b_{.\ell}}$ or the minimization of the loss in mutual information due to co-clustering, i.e,

$$\sum_{b=1}^{v} \sum_{i=1}^{n} \sum_{j=1}^{d} p^b_{ij} \log \frac{p^b_{ij}}{p^a_{i.} p^b_{.j}} - \sum_{b=1}^{v} \sum_{k=1}^{g} \sum_{\ell=1}^{m} p^b_{k\ell} \log \frac{p^b_{k\ell}}{p^b_{k.} p^b_{.\ell}}. \tag{2.11}$$

Note that for $v = 1$, (2.11) is the objective function optimized by ITCC [Dhillon et al., 2003] or the Croinfo algorithm [Role et al., 2019]. Hence, the CEM−T algorithm can be viewed as a model-based clustering version of ITCC/Croinfo where the proportions of row clusters (resp.

column clusters) are assumed to be equal; see for instance [Govaert and Nadif, 2018, Ailem et al., 2017].

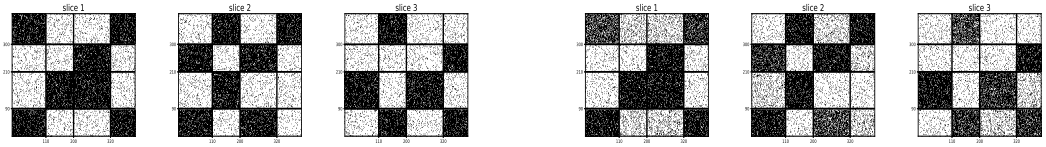
2.5 Experimental results

The evaluation of co-clustering is generally carried out the basis on benchmarks datastets where only one of the two partitions is known. In the same way we compare VEM-T with competitive (co)-clustering methods. We retain three widely used measures to assess the quality of clustering, namely the accuracy, the Normalized Mutual Information (NMI) [Strehl and Ghosh, 2002] and the Adjusted Rand Index (ARI) [Liu et al., 2013b].

We present results on real datasets for three different areas namely recommender systems, multi-spectral images clustering and documents categorization. Through this evaluation, we aim to demonstrate the impact of covariate information on interpretation and improvement of clustering results.

2.5.1 Synthetic datasets and Competitive methods

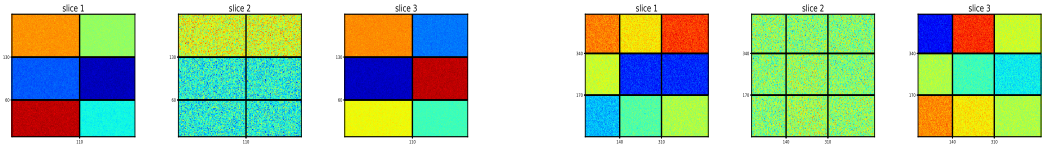
Before proceeding to evaluate VEM-T on real datasets, we give here two simple illustrative examples. We generated tensor data $\mathcal{X}$ according to the Bernoulli and Gaussian TLBM (Algorithm 3) with $v = 3$. Following each model, we considered two scenarios by varying the centers $\boldsymbol{\mu}_{k\ell}$'s; an example where the co-clusters are well separated and another where the co-clusters are not. The size of each tensor, number of co-clusters and their proportions are reported in Tables 2.2,2.3. Herein other characteristics of each tensor dataset: continuous data we take the same covariance matrix for all blocks $\begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$ for example 3 and $\begin{bmatrix} 1 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 \\ 0.8 & 0.8 & 1 \end{bmatrix}$ for example 4. All variables (slice) are standardized to have values between zero and one. In Figures 2.5 and 2.4 are depicted the true simulated tensor data into $v = 3$ slices.



Example 1 Example 2

FIGURE 2.4: Simulated binary datasets.

In our experiments, we compare VEM-T with K-means, Gaussian Mixture Model (EMGMM: EM with the full model, see for instance [Fraley and Raftery, 1998]) and VEM for co-clustering applied on each slice [Govaert and Nadif, 2006]. The NMI metric for rows and columns are computed by averaging on ten random initializations. Thereby, in Tables 2.2 and 2.3 are reported the performances for the three slices obtained by K-means, EMGMM, VEM for data matrix and by VEM-T for tensor data. From these comparisons, we observe that whether the



Example 3 Example 4

FIGURE 2.5: Simulated continuous datasets.

TABLE 2.2: Evaluation of co-clustering in terms of NMI for binary datasets.

Algorithm	Metrics		Example 1 $400 \times 400 \times 3$ $(g,m) = (4,4)$ $\pi = [0.23, 0.3, 0.23, 0.24]$ $\rho = [0.27, 0.23, 0.3, 0.2]$			Example 2 $400 \times 400 \times 3$ $(g,m) = (4,4)$ $\pi = [0.23, 0.3, 0.23, 0.24]$ $\rho = [0.27, 0.23, 0.3, 0.2]$		
	NMI		Slice 1	Slice 2	Slice 3	Slice 1	Slice 2	Slice 3
K-means	NMI	Row	0.80 ± 0.00	0.81 ± 0.00	0.98 ± 0.00	0.83 ± 0.00	0.82 ± 0.00	0.94 ± 0.00
		Column	0.83 ± 0.01	0.76 ± 0.00	0.76 ± 0.00	0.83 ± 0.012	0.80 ± 0.00	0.80 ± 0.00
EMGMM	NMI	Row	0.81 ± 0.00	0.81 ± 0.00	1.00 ± 0.00	0.82 ± 0.00	0.82 ± 0.00	0.90 ± 0.02
		Column	0.79 ± 0.02	0.77 ± 0.01	0.76 ± 0.01	0.83 ± 0.01	0.88 ± 0.00	0.83 ± 0.01
VEM	NMI	Row	0.66 ± 0.00	0.72 ± 0.00	0.78 ± 0.01	0.71 ± 0.00	0.73 ± 0.00	0.86 ± 0.01
		Column	0.70 ± 0.00	0.71 ± 0.00	0.71 ± 0.00	0.72 ± 0.00	0.73 ± 0.00	0.80 ± 0.00
VEM-T	NMI	Row		**0.94 ± 0.00**			**0.90 ± 0.00**	
		Column		**0.93 ± 0.00**			**0.97 ± 0.00**	

TABLE 2.3: Evaluation of co-clustering in terms of NMI for continuous datasets.

Algorithm	Metrics		Example 3 $200 \times 200 \times 3$ $(g,m) = (3,2)$ $\pi = [0.3, 0.35, 0.35]$ $\rho = [0.55, 0.45]$			Example 4 $500 \times 500 \times 3$ $(g,m) = (3,3)$ $\pi = [0.34, 0.34, 0.32]$ $\rho = [0.28, 0.34, 0.38]$		
	NMI		Slice 1	Slice 2	Slice 3	Slice 1	Slice 2	Slice 3
K-means	NMI	Row	1.0 ± 0.0	0.62 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.09 ± 0.0	1.0 ± 0.0
		Column	1.0 ± 0.0	0.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.29 ± 0.01	1.0 ± 0.0
EMGMM	NMI	Row	1.0 ± 0.0	0.62 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.15 ± 0.0	1.0 ± 0.00
		Column	1.0 ± 0.0	0.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.24 ± 0.01	1.0 ± 0.00
VEM	NMI	Row	0.98 ± 0.0	0.77 ± 0.0	0.98 ± 0.0	0.95 ± 0.01	0.50 ± 0.0	1.0 ± 0.00
		Column	1.0 ± 0.0	0.0 ± 0.0	1.0 ± 0.0	0.95 ± 0.01	0.60 ± 0.0	0.95 ± 0.01
VEM-T	NMI	Row		**1.0 ± 0.00**			**0.95 ± 0.01**	
		Column		**1.0 ± 0.00**			**0.95 ± 0.00**	

block structure is easy to identify (Examples 1,3) or not (Examples 2,4), the ability of VEM-T to outperform other algorithms that, it should be recalled, act on each slice separately.

2.5.2 Real datasets

We tested the performance and felxibility of the proposed models using tensor real-world datasets. In particular, we focused on binary, continuous and count data, with different applications including Recommender system, Multi-spectral images clustering and Document clustering. The characteristics of these tensor datasets are summarized in table 2.4.

TABLE 2.4: Characteristics of datasets.

Application	Datasets	#Tensor mode-1	#Tensor mode-2	#Tensor mode-3	Sparsity
Recommender system	Movielens-100K	943	1682	42	0.93
Multi-spectral images clustering	Prostate-Cells	37	16	14	0.
Document clustering	DBLP1	2223	2223	4	0 .93
	DBLP2	1949	1949	4	0.94
	PubMed-Diabets	4354	4354	4	0.69

Recommender system application

To show the benefits of our approach, we use the binary model on Movielens100K which is one of the more popular datasets on the recommender system field. The objective of this study is identifying patterns according to users and movies characteristics. The Movielens100K[1] database consists of 100,000 ratings of 943 users and 1682 movies, where each user has rated at least 20 movies. We convert the users-movies rating matrix (943×1682) to binary matrix by assigning 0 to the movie without rating and 1 to rated movies. This binary matrix can be considered as viewing matrix, in fact most users rates movies after watching them. Furthermore, Movielens includes 22 user covariates including age, gender, and 21 employment status. The age covariate is used to analyze clustering results and does not take into account in co-clustering. There are also 19 movie covariates related to movie genres, considering that movie may belong to one or more genres. The data structure can be represented as tensor with size $943 \times 1682 \times 42$. The objective of this work is not being to select the number of clusters, then we fixed the number of row clusters $g = 2$ and the number of column clusters $m = 3$, based on the works of [Vu and Aitkin, 2015]. Figures 2.6 and 2.7a represent the mean vectors $\boldsymbol{\mu}_{k\ell}$ and co-clustering of rating matrix respectively. We observe two row clusters, a smaller cluster of 202 users which is more active in reviewing than a second large cluster. On the other hand, we obtain three movies clusters of different sizes 232, 355 and 1,095 respectively. The first cluster represents the most attractive movies.

The first row cluster includes three blocks (1,1), (1,2) and (1,3). The two first ones represent the more active users with a higher proportion of rating. The MovieLens100K dataset includes 29% of female reviews, an important part of them (64%) belong to a first row cluster. In addition, we notice that the top 3 of occupations for users of the first row cluster are a student, educator, and administrator. Thereby, Figure 2.7b shows that 65% of them are quite young and under 31 years of age. However, the two blocks (1,1) and (1,2) are distinguished by movie genres, since the top 3 ones for first and second column clusters are Action-Thriller-Sci-Fi and Comedy-Drama-Romance respectively. Consequently, we can

[1] http://grouplens.org/datasets/movielens/

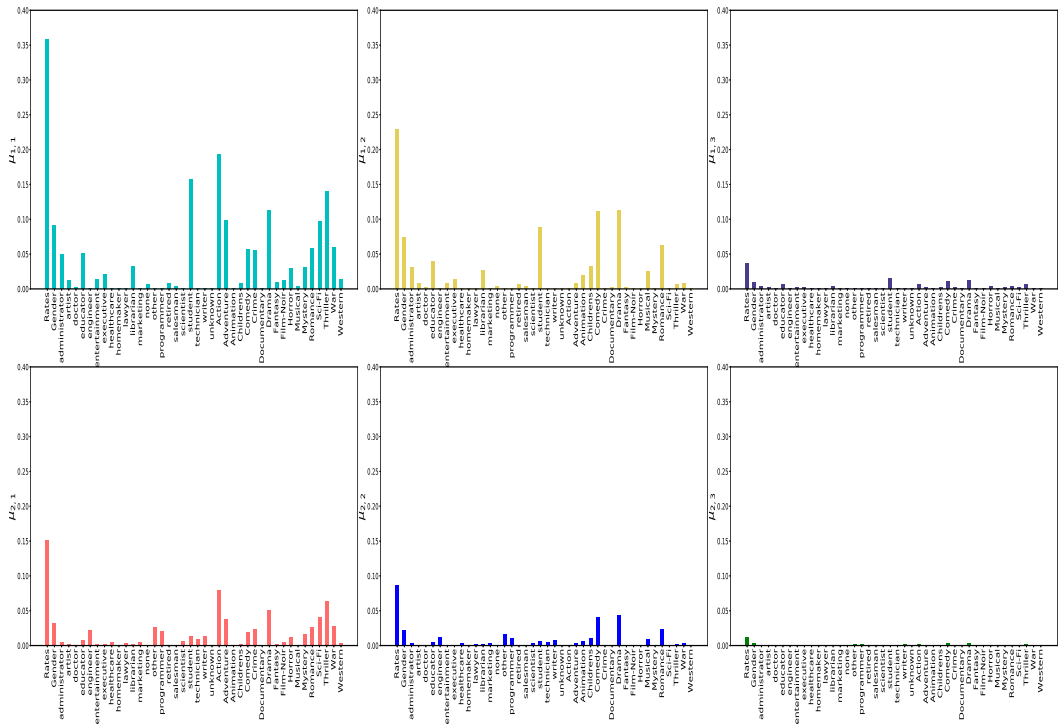
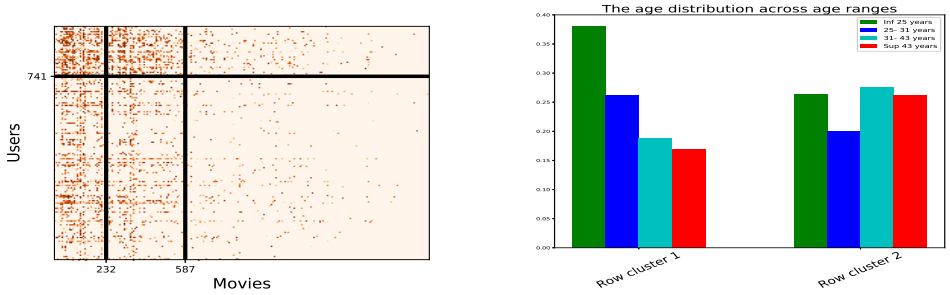


FIGURE 2.6: Distribution of the centers $\mu_{k\ell}$ for all co-clusters.

identify two profiles of young active users; they are attracted by both categories of movies namely Action-Thriller-Sci-Fi for the first profile and Comedy-Drama-Romance for the second. The second row cluster regroups the users of different ranges of age with almost equal proportions (see Figure 2.7.b) and different occupations since the top three occupations include engineer, student, and another employment status. Finally the third column cluster seems representing movies with different genres Action-Drama-comedy. The block (2,3) represents the less attractive movies watched by the less active users.



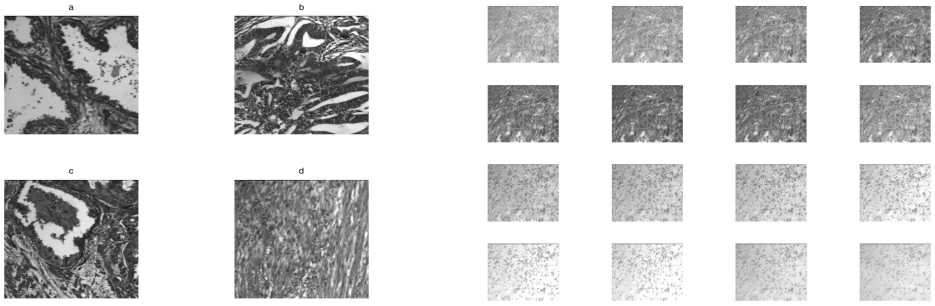
(a) (b)

FIGURE 2.7: (a) Co-clustering data matrix, (b) Distribution of Age per row clusters.

Multi-spectral images analysis

The used dataset is composed by 37 multispectral images of prostate cells with 16 bands which have size 512×512 pixels. Several studies showed that clustering accuracy increases according bands number [Kumar and Sreekumar, 2014]. The four types of multispectral images cells are: Normal cells (Stroma), Benign Hyperplasia (BHp), Interpithelial Neoplasy (PIN) which is a cancer precursory state, and the Carcinoma (CA) which corresponds to a cancer of the abnormal tissue proliferation. Figure 2.8.a presents cell's types and the example of 16 bands of Stroma cells type are showed in figure 2.8.b. Some elements allow to differentiate the cell's types, among those morphological and textural features. In this way, we limited ourselves to textural characteristics for clustering. Haralick [Haralick et al., 1973] defined several metrics computed from the gray level co-occurrence matrix (GLCM). The Haralick's parameters showed their efficiency in the literature for the textures analysis [Haralick et al., 1973, Kumar and Sreekumar, 2014]. The 14 Haralick's features are the following: Energy, Correlation, Contrast, Entropy, Homogeneity, Inverse Difference Moment, Sum Average, Sum Variance, Sum Entropy, Difference Average, Difference Variance, Difference Entropy and two Information measure of correlation.

In the most previous studies, the extraction of 14 Haralick's features from all bands are performed, and the 14×16 features are extracted for each image involving features selection or dimensionality reduction with popular methods such as PCA. These operations can provide interesting results but leads to a loss of information. To overcome this drawback, we propose to construct tensor data $Images \times Bands \times Features$ in order to exploit all available data without requiring dimensionality reduction. The objective of this study is improving clustering results of multispectral images which highly used on biomedical and geology fields.



(a) (b)

FIGURE 2.8: (a) The four cells type, (b) Example of multispectral image from dataset.

As we known the true number of image clusters, we take $g = 4$ and as we have no information about column clusters we postulate $m = g = 4$. As shown in Figure 2.9, the Stroma cells are characterized by higher values of entropy, contrast and difference variance on the first three column clusters, and low values of inverse difference moment feature on two first band clusters. The PIN type is characterized by low values of information measure correlation 1 on bands cluster 2, 3 and 4. The cell type with the closer values of features is BHP. The CA type is characterized by higher values of information measure correlation 1 on the third and fourth band clusters and the lower values of information measure correlation 2

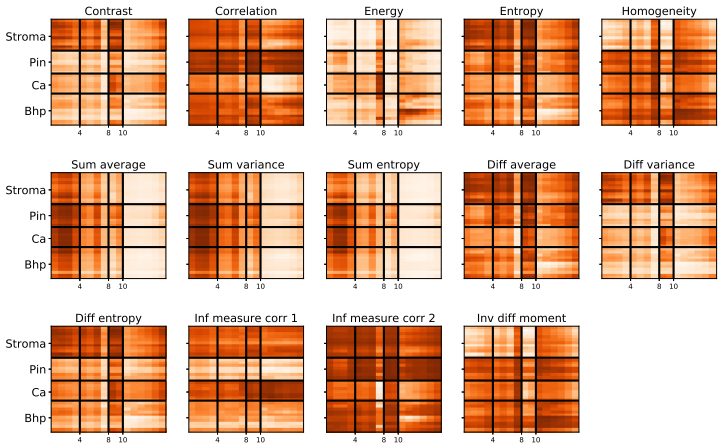


FIGURE 2.9: Co-clustering matrix of different slice of features.

on all bands. Finally, BHP cells are characterized by the lowest values of sum average on two last bands clusters.

The VEM-T algorithm is compared with K-means, EMGMM and VEM. For this, a reduced matrix of tensor data by averaging all bands for each feature provides a *Images* $\times$ *Features* data matrix used to perform classical clustering. Table 2.5 summarizes the obtained results. For each algorithm, the best result rather than 100 random initial runs are used. Clearly the proposed algorithm achieves best results as regards NMI, ARI and ACC (Accuracy).

TABLE 2.5: Evaluation of K-means, EMGMM, VEM and VEM-T in terms of NMI, ARI and ACC.

Algorithms	NMI	ARI	ACC
K-means	0.67	0.56	0.78
EMGMM	0.7	0.59	0.78
VEM	0.61	0.49	0.7
VEM-T	**0.90**	**0.87**	**0.95**

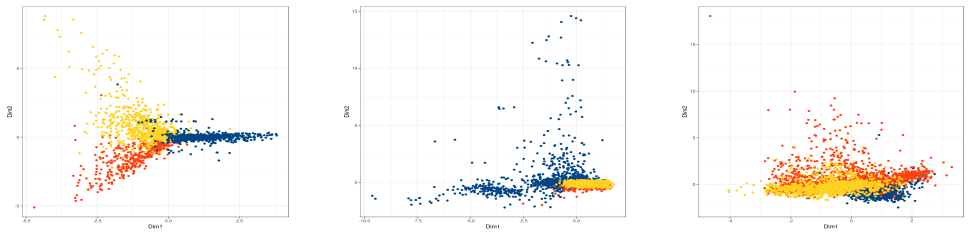
Document categorization

In our experiments, we aim to evaluate VEM-T for contingency tables in terms of clustering leading to measure the impact of mixing different information. Thereby, we compare VEM-T with Spherical K-means, Itcc [Dhillon et al., 2003], and VEM-T$_b$ applied on each slice (b) of tensor and three other algorithms applied on tensor data namely PARAFAC [Kossaifi et al., 2019] and GTSC [Wu et al., 2016]. Note that PARAFAC is used with ranks number equals to 10 and followed by K-means. We perform 50 random initializations, and compute the ACC, ARI and NMI metrics by averaging the ten top runs.

We use three text datasets DBLP1, DBLP2 and PubMed Diabetes[2] to highlight the objective of the proposed algorithm. DBLP1 and DBLP2 are constructed from DBLP[3], by

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

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.



DBLP1 DBLP2 PubMed Diabets

FIGURE 2.10: Obtained results using Multiple Factor Analysis.

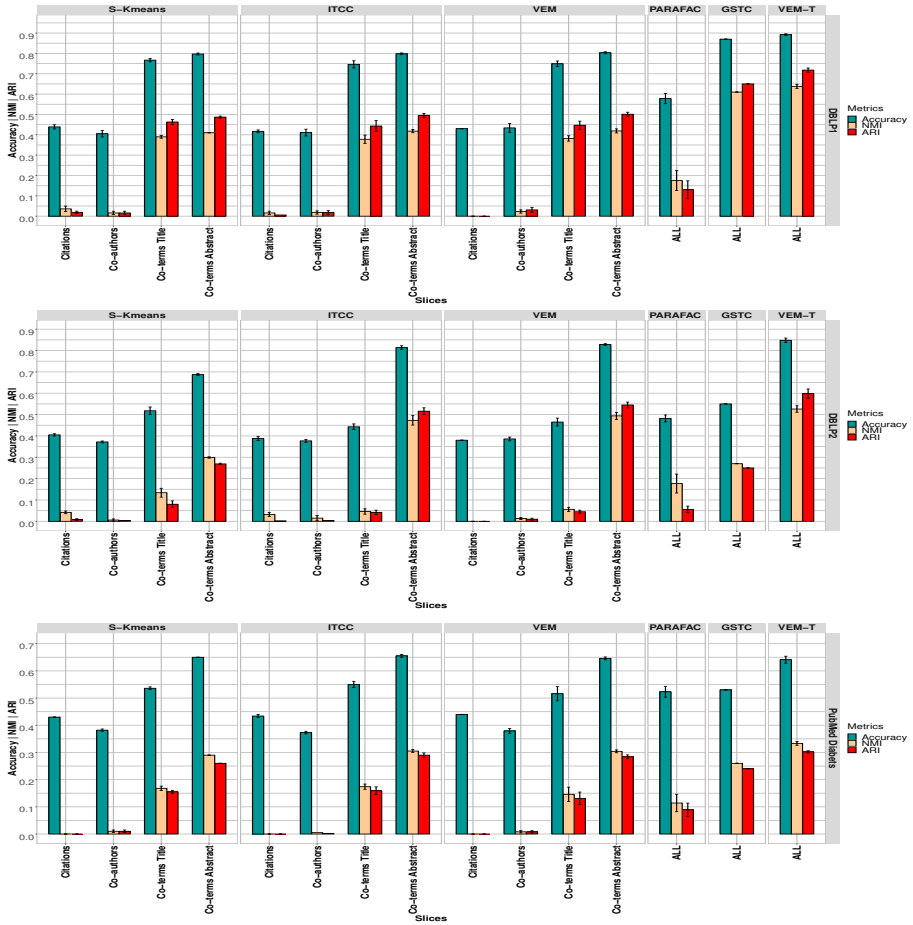
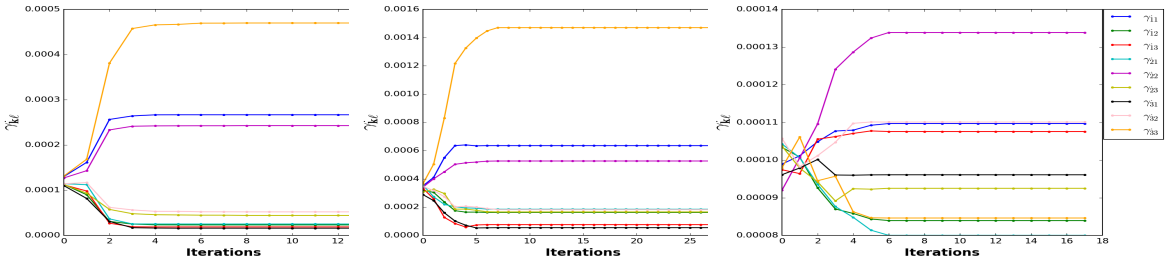


FIGURE 2.11: Obtained results on DBLP1, DBLP2 and PubMed datasets using `S-Kmeans`, `ITCC`, `VEM`, `PARAFAC`, `GSTC` and `VEM-T`.

From these different datasets, we construct the following adjacency matrices:

- 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. This matrice is computed using $\mathcal{T}\mathcal{T}^T$ where $\mathcal{T}$ is a binarized documents-terms matrix.
- Co-terms matrix on the abstract: each cell represents the number of times that a pair of papers share a term extracting from abstract. We use the same process that used in *Co-terms Title matrix*.
- Co-authors matrix: each cell represents the number of common authors for a pair of papers. This matrice is computed using $\mathcal{A}\mathcal{A}^T$ where $\mathcal{A}$ is a binarized documents-authors matrix.
- 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.



DBLP1 DBLP2 PubMed Diabets

FIGURE 2.12: Behavior of the γ_{kl} parameters at each iteration.

Plots figure 2.10 represent the low-dimensional projection of papers from tensor data of DBLP1, DBLP2 and PubMed Diabetes respectively using the *Multiple Factor Analysis* (MFA). MFA deals with a multiple table where the slices are contingency tables [Pagès, 2014]. We notice that the three datasets have different degree of complexity.

In figure 2.11 are reported the performances of the six algorithms (cited above) on the three datasets. In terms of ACC, NMI and ARI, we observe in most cases, that `VEM-T` is better than other algorithms applied on each slice and those applied on tensor data. With 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 2.10. Further note that `GTSC`, less effective than `VEM-T`, reaches better results than `PARAFAC` followed by `K-means`. We can notice that $\texttt{VEM-T}_b$ applied on each slice does a good job on the well-separated slices like co-terms Title and co-terms Abstract. Finally, we can say that the `VEM-T` with considering all slices (the well-separated one and the ill-separated one) can find the best trade-off in terms of clustering results.

Furthermore, Figure 2.12 shows the behaviour of the $\gamma_{k\ell}^{[.]}$ parameter for each block at each iteration, for the three datasets. $\gamma_{k\ell}^{[.]}$ is computed at each iteration by averaging all $\gamma_{k\ell}^{b}$ as

$\gamma_{k\ell}^{[.]} = \frac{1}{v}\sum_{b=1}^{v}\gamma_{k\ell}^{b}$. Interestingly, for DBLP1 and DBLP2, we can see that while the average of diagonal parameters $\gamma_{kk}^{[.]}$ increases, the value of the parameter $\gamma_{k\ell}^{[.]}$ where $k \neq \ell$, decreases at each iteration. For these two datasets, we have three well-separated clusters on diagonal which explain that $\gamma_{kk}^{[.]}$ increases perfectly and $\gamma_{k\ell}^{[.]}$ where $k \neq \ell$ also decreases perfectly. For PubMed Diabets, the data structure seems more complicated, and then the interpretation of gamma evolution is more complicated. We can see that $\gamma_{k\ell}^{[.]}$ increases for four blocks and decreases otherwise.

As we have seen CEM−T is a hard version of VEM−T, what is then its behavior in terms of computational time and clustering performance? Thereby, in figure 2.13 we report the comparison between CEM−T and VEM−T with the three datastets. The CEM−T algorithm is faster than VEM−T but in terms of clustering performances (Accuracy, NMI and ARI), we can see that VEM−T is at least equivalent.

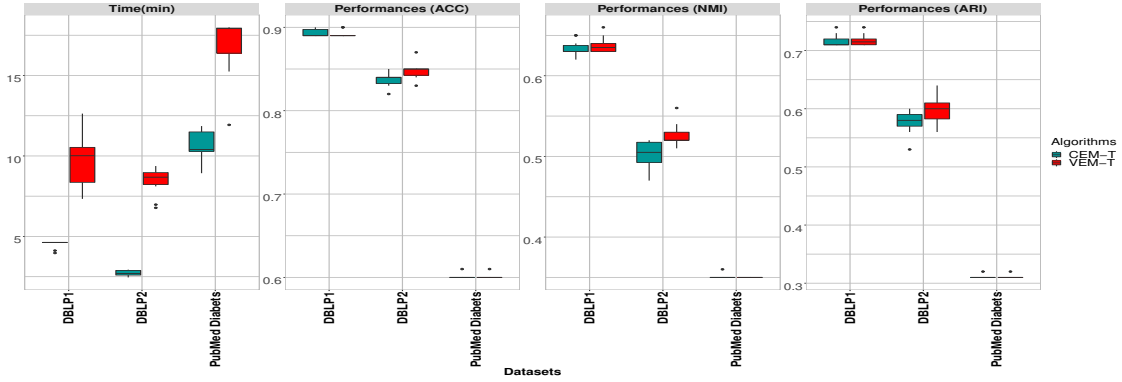


FIGURE 2.13: Comparaison of CEM−T and VEM−T in terms of time complexity and performances.

2.6 Conclusion

Inspired by the flexibility of the latent block model (LBM), in this chapter we proposed a tensor version of LBM (TLBM). This given rise to new variational EM algorithm for co-clustering of different types of data. Empirical results on synthetic and real-world datasets – binary, continuous, and contingency tables– showed that VEM−T and its hard version CEM−T do a better job than other algorithms devoted to the same task or other algorithms applied on each slice of tensor data. Furthermore, we have shown that VEM−T is efficent for several applications, namely the recommender system, hyperspectral image clustering, and document categorization. More interestingly, our findings open up good opportunities for future research such as the analysis of temporal data or assessing the number of co-clusters.

Chapter 3

Sparse Poisson Tensor Co-clustering

3.1 Introduction

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. In fact, 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 in 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` [Tucker, 1966] and `PARAFAC` [Harshman and Lundy, 1994]. We can nevertheless mention the works related to our proposal, such as the work of [Banerjee et al., 2005] based on Minimum Bregman information (`MBI`) to find co-clustering of a tensor. Most recently, in [Wu et al., 2016] the General Tensor Spectral Co-clustering (`GTSC`) method for co-clustering the modes of non-negative tensor has been developed. In [Feizi et al., 2017], the authors proposed a tensor biclustering algorithm able to extract the most important bi-cluster based on spectral decomposition and the obtained eigenvalues and offer an application for microarray analysis. 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.

In this work, we offer a generalized model for co-clustering Tensor Sparse PLBM (TSPLBM) dealing with sparse tensor with different mode size. The goal is to simultaneously discover row and column clusters and the relationship between these clusters for all slices. Then a particular case for semi-symmetric tensor (with the same size for the two first mode) is proposed. We illustrate the interest of this model with application to the clustering of multiple graphs. Also, the TSPLBM model for sparse tensor data 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.

To the best of our knowledge, this is the first attempt to formulate our objective when both sets of first and second modes can be different and with model-based co-clustering. To this end, we rely on the latent block model [Govaert and Nadif, 2013] for its flexibility to consider any data matrices. The key contributions of this work are:

- We first develop a novel TSPLBM model for the co-clustering of sparse tensor data, composed by multiple contingency tables.

- We show the links between *Poisson Latent Block Model* (PLBM) and the *Poisson Stochastic Block Model* (PSBM). Then we discuss the strong points of PLBM, PSBM and SPLBM (Sparse Poisson Latent Block Model) in terms of graph clustering.
- We propose a suitable probabilistic model for clustering of multiple graphs, then we derive an EM-type learning algorithm.
- Finally, using the ensemble method, we prove that the proposed algorithm, which can be viewed as an implicit consensus clustering for multiple graphs, is more effective than explicit clustering obtained by consensus clustering methods.

The remainder of this chapter is organized as follows. In Section 3.2, we present a sparse tensor co-clustering model TSPLBM. Section 3.3 reviews Poisson LBM, shows the limits of traditional PSBM, and adapt TSPLBM for multiple graphs. Section 3.4, is devoted to evaluating our approach and demonstrate the strong points of implicit consensus trough TSPLBM and explicit consensus methods. Finally, section 3.5 concludes the chapter and gives some directions for future works.

3.2 Sparse Tensor Co-Clustering

In this section, we will detail the Sparse Poisson Latent Block Model (SPLBM) and gives the intuition behind the model and their parameters. After that, we present our extension of SPLBM for tensor data, which is Sparse Tensor PLBM (or STPLBM). The suitable Variational EM algorithm is derived (VEM-ST) and presented at the end of this section.

3.2.1 Sparse Poisson LBM (SPLBM)

Despite the effective parameterization of the Poisson LBM (see sections 2.2.1 and 2.2.2, it remains insufficient because it suffers from sparsity.

Recently, in [Ailem et al., 2017], the authors proposed a generative mixture model for co-clustering document-term matrices referred to as SPLBM. With this model, they assume that for each diagonal block kk the values $x_{ij} \sim \text{Poisson}(\lambda_{ij})$ where

$$\lambda_{ij} = x_{i.}x_{.j}\sum_{k}[z_{ik}w_{jk}]\gamma_{kk} \quad or \quad x_{ij}|z_{ik}w_{jk} = 1 \sim \mathcal{P}(x_{i.}x_{.j}\gamma_{kk}),$$

and for each block $k\ell$ with $k \neq \ell$, $x_{ij} \sim \text{Poisson}(\lambda_{ij})$ where the parameter λ_{ij} takes the following form:

$$\lambda_{ij} = x_{i.}x_{.j}\sum_{k,\ell\neq k}[z_{ik}w_{j\ell}]\gamma \quad or \quad x_{ij}|z_{ik}w_{j\ell} = 1 \sim \mathcal{P}(x_{i.}x_{.j}\gamma).$$

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. The pdf of SPLBM can be

written as follows:

$$f(\mathbf{X},\mathbf{\Omega}) = \sum_{(\mathbf{Z},\mathbf{W})\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,\ell} \rho_\ell^{w_{j\ell}} \prod_{i,j,k} \big(\Phi(x_{ij};\lambda_{kk})\big)^{z_{ik}w_{jk}} \prod_{i,j,k,\ell\neq k} \big(\Phi(x_{ij};\lambda)\big)^{z_{ik}w_{jk}}.$$

Assuming that the complete data are $(\mathbf{X},\mathbf{Z},\mathbf{W})$, the complete data log-likelihood $L_C(\mathbf{Z},\mathbf{W},\mathbf{\Omega})$ takes the following form :

$$\log\left(\prod_{i,k}\pi_k^{z_{ik}}\prod_{j,\ell}\rho_\ell^{w_{j\ell}}\prod_{i,j,k}\left(\frac{e^{-x_{i.}x_{.j}\gamma_{kk}}(x_{i.}x_{.j}\gamma_{kk})^{x_{ij}}}{x_{ij}!}\right)^{z_{ik}w_{jk}}\prod_{i,j,k,\ell\neq k}\left(\frac{e^{-x_{i.}x_{.j}\gamma}(x_{i.}x_{.j}\gamma)^{x_{ij}}}{x_{ij}!}\right)^{z_{ik}w_{jk}}\right).$$

To estimate the parameters $\mathbf{\Omega}$, $\mathbf{Z}$ and $\mathbf{W}$. To this end, a variationnel EM has been proposed [Ailem et al., 2017] to maximize (2.8) where $L_C(\widetilde{\mathbf{Z}},\widetilde{\mathbf{W}},\mathbf{\Omega})$ is the new fuzzy complete-data log-likelihood.

Note that plugging the estimation of γ_{kk}'s and γ (explicitly in some terms of L_C) deduced from the maximization step, we obtain

$$\begin{aligned} L_C(\mathbf{Z},\mathbf{W},\mathbf{\Omega}) = &\sum_{i,k}\tilde{z}_{ik}\log\pi_k + \sum_{j,k}\tilde{w}_{jk}\log\rho_k \\ &+\left(\sum_k\left[x_{kk}\log(\frac{\gamma_{kk}}{\gamma}) - x_{k.}x_{.k}(\gamma_{kk}-\gamma)\right] + N\log(\gamma) - N^2\gamma\right). \end{aligned}$$

Then the computation of z_{ik}, $w_{j\ell}$ and the parameters $\mathbf{\Omega} = (\boldsymbol{\pi},\rho,\gamma_{kk},\gamma)$ can be easily deduced from the derivation $L_C(\mathbf{Z},\mathbf{W},\mathbf{\Omega})$.

Note that although SPLBM is a co-clustering model, we can derive a graph clustering algorithm from an adjacency matrix (symmetric or not). Thereby, when we are dealing with undirected graphs; strating with the same initialization of $\mathbf{z}$ and $\mathbf{w}$ ($\mathbf{z}^{(0)} = \mathbf{w}^{(0)}$), we obtain the same row and column clusters, that is essential for the undirected graph clustering problem.

Although PLBM can deal with sparse matrices, SPLBM can be more suitable for sparse matrices (see figure 3.1). It is designed to seek a diagonal block structure and capture the most reliable associations between the rows and columns object clusters. SPLBM assumes that each diagonal block (or co-cluster) is generated according to the Poisson distribution with some specific parameters, and each non-diagonal co-cluster representing noise data is generated according to Poisson distribution with identical parameters.

3.2.2 Tensor Sparse Poisson LBM (TSPLBM)

Tensor LBM (TLBM) is a novel Latent Block Model based on multivariate distribution (see section 3). While the traditional Latent Block Model (LBM for co-clustering) seeks to discover homogeneous blocks modeled by univariate distribution, TLBM can deal with multi-view data structured as a three-way tensor.

In this work, we extend the SPLBM to Tensor data leading to Tensor SPLBM (or TSPLBM). The proposed model seeks not only to discover homogeneous tube co-clusters but also discover important blocks and ignore noisy ones. TSPLBM aims to discover a diagonal co-clusters structure, which is tubes (trough all slices) from the three-way tensor. It makes it more useful for sparse tensor with high sparsity close to 90%, as shown in the experiments.

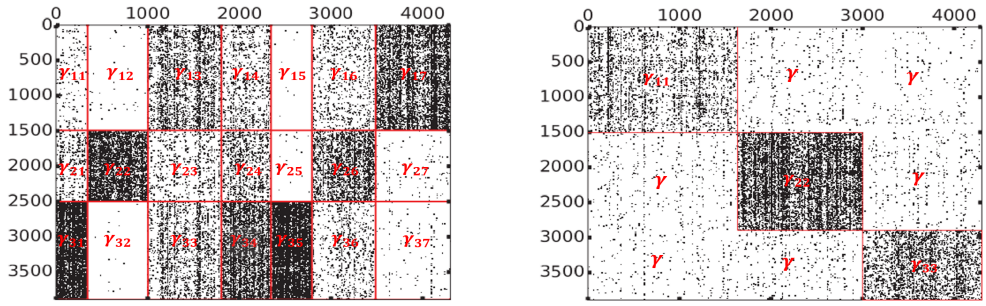


FIGURE 3.1: Difference between PLBM and SPLBM paramertization.

TSPLM provides a better partitioning than applying the classical co-clustering algorithm on each slice of tensor separately and using a consensus clustering on these independent results. The PDF function of the proposed TSPLBM can be written as follows:

$$\sum_{(\mathbf{z},\mathbf{w})\in\mathcal{Z}\times\mathcal{W}} \prod_{ik} \pi_k^{z_{ik}} \prod_{j\ell} \rho_\ell^{w_{j\ell}} \prod_{i,j,k} \left(\prod_{b=1}^{v} \Phi(x_{ij}^b; \lambda_{kk}^b) \right)^{z_{ik}w_{jk}} \prod_{i,j,k,\ell\neq k} \left(\prod_{b=1}^{v} \Phi(x_{ij}^b; \lambda^b) \right)^{z_{ik}w_{jk}}.$$

In the following we propose to extend SPLBM to deal with tensor data. The log-likelihhod of TSPLBM $L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega})$ takes the following form:

$$\begin{aligned} &\sum_{i,k} \tilde{z}_{ik} \log \pi_k + \sum_{j,k} \tilde{w}_{jk} \log \rho_k \\ &+ \sum_{i,j,k} \tilde{z}_{ik}\tilde{w}_{jk} \left(\sum_{b=1}^{v} \log \Phi(x_{ij}^b; \lambda_{kk}^b) \right) + \sum_{i,j,k,\ell\neq k} \tilde{z}_{ik}\tilde{w}_{j\ell} \left(\sum_{b=1}^{v} \log \Phi(x_{ij}^b; \lambda^b) \right). \end{aligned} \tag{3.1}$$

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, the log-likelihood expression in equation (3.1) becomes (up a constant)

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

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 $N_b = \sum_{i,j} x_{ij}^b$.

3.2.3 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 $\mathbf{\Omega}$. 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 (See Appendix A)

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

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

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, γ_{kk}^b and γ^b maximizing $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega})$. 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 (See Appendix C)

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

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

The proposed algorithm for sparse tensor (ST) data, referred to as `VEM-ST` in Algorithm 6, 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 6: `VEM-ST`

Input: $\mathcal{X}$, g.
Initialization $(\mathbf{Z}, \mathbf{W})$ randomly, compute $\mathbf{\Omega}$
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 $\mathbf{\Omega}$
until *convergence*;
return Ω, Z, W

3.3 Clustering from Multiple Graphs

Relational data are ubiquitous in various fields (web, biology, neurology, sociology, communication, economics, etc.), and their accessibility has kept increasing in recent years. These data, as a whole, form a network formalized by a graph, where each node is an entity, and each edge is a connection between a pair of nodes; this graph can be directed or not. We find this situation in various scientific publications; the relationships between documents can often be described as multiple graphs with different types of links. In fact, several relationships, such as co-terms, co-authors, co-keywords, and co-references between documents can be used. The objective of this work is to address the clustering of multiple graphs. We could hypothesize that the combination of different information that arises from multiple graphs may improve the clustering results. In fact, two documents which share a number of words and/or have one or more authors in common and/or 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.

To deal with multiple graphs, various models and methods under different approaches are proposed to analyze these networks. In [Banerjee et al., 2007, Tang et al., 2009], the authors proposed a multi-way clustering framework for relational data, where different types of entities are simultaneously clustered, based not only on their intrinsic attribute values, but also on the multiple relations between the entities. Other works use a spectral decomposition-based approach relying on the combination of adjacency matrices [Tang et al., 2009, Chen et al., 2017, Nie et al., 2017]. In these works, the clustering is not the main objective of the proposed approaches, nevertheless it can be deduced from decomposition results.

On the other hand, one of the most used methods in this context is the *Stochastic Block Model* (SBM) [Nowicki and Snijders, 2001] which is a probabilistic approach. SBM is commonly used for network modeling and discovering the latent community structures from a graph. It provides a statistical approach able to model data matrix, symmetric or not, into homogeneous blocks. This leads to consider SBM [Daudin et al., 2008] as a particular case of the *Latent Block Model* (LBM) [Govaert and Nadif, 2003, Govaert and Nadif, 2005] and extended in [Shan and Banerjee, 2008, Govaert and Nadif, 2013], which models any kind of data matrices not necessarily square or symmetric. In other words, the clustering of the graph directed or not, is in fact, a particular case of co-clustering. In this work, we consider graphs represented by adjacency matrices assimilated to contingency tables. Thus, considering the previous example of document clustering, the relations between documents (co-terms, co-authors, etc.) are count data and can be represented by particularly sparse contingency tables. Many works in the literature show the interest of Poisson distribution for graph theory and clustering of random graphs [Janson, 1987, Daudin et al., 2008].

In this section, we adapt the previously proposed TSPLBM to the clustering of multiple graphs. For this aim, we present a special case of TSPLBM dealing with semi-symmetric tensor (see section 1.3.1).

Our current contribution significantly expands the applicability of the model-based co-clustering framework. Specifically, based on LBM, the contribution proposes (a) a novel version of the Poisson SBM (PSBM) for multiple graphs (b) a simultaneous co-clustering of multiple graphs leading to a kind of consensus clustering. Figure 3.2 presents a binary three-way dataset constructed from multiple graphs and the expected results in terms of co-clustering.

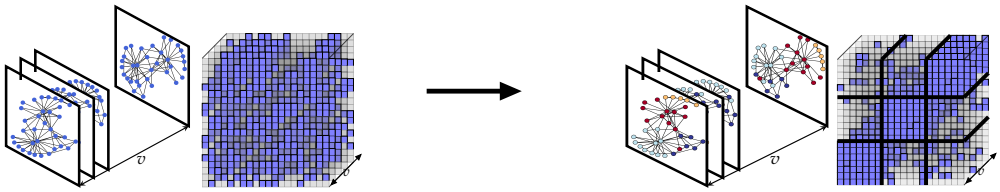


FIGURE 3.2: Goal of co-clustering of multiple graphs.

3.3.1 Related Work

Although SBM is popular in social networks analysis, dealing with the count data and due to the degree of heterogeneity, the traditional SBM fail to detect relevant clusters of edges to adress community detection problem [Qiao et al., 2017]. Thereby, several authors have developed a degree-corrected SBM. In [Karrer and Newman, 2011], using a Poisson SBM, they introduced a parameter θ_i controlling the degree of expected degrees of vertices i. They consider that each x_{ij} with $i \neq j$ is distributed according to Poisson$(\theta_i\theta_j\delta_{k\ell})$, where $\delta_{k\ell}$ is the expected value of the adjacency matrix for the vertices i and j lying in block (k,ℓ) while x_{ii} is distributed according to Poisson$(\frac{1}{2}\theta_i^2\delta_{kk})$. Doing so and under some constraints on the θ_i's, they proposed the DC-SBM (Degree-Corrected SBM) clustering algorithm (`DC-SBM`[1]) from an undirected graph on n vertices, possibly including self-edges. Furthermore, they established the equivalence between the maximization of the log-likelihood and the maximization of mutual information used as an objective function for clustering bipartite graphs [Dhillon et al., 2003]. It is important to emphasize that the model proposed in [Karrer and Newman, 2011] is similar to that proposed by [Nadif and Govaert, 2005], where the authors also showed this connection with the maximization of mutual information; they proposed the `Croinfo` algorithm as illustrated in Figure 3.3. In fact, the objective function maximized by `DC-SBM`, which can also be used for the co-clustering of an undirected graph, is associated with a *constrained* Poisson LBM commonly used in the co-clustering context; see e.g.; [Ailem et al., 2017, Ailem et al., 2017]. To sum up, considering `DC-SBM` which implies that the data are generated according to a Poisson LBM with $\mathcal{P}(x_{ij}, x_{i.}x_{.j}\gamma_{k\ell})$ where $\mathcal{P}(x_{ij};\lambda) = \frac{e^{-\lambda}\lambda^{x_{ij}}}{x_{ij}!}$, the proportions of the classes of the nodes are assumed to be equal. In addition, although both algorithms `DC-SBM` or `Croinfo` are different, the objective is the same, and the clustering considered is based on an approach similar to that of the traditional hard clustering algorithms; for more detail, the reader can refer to recent works [Govaert and Nadif, 2013, 2018].

In our contribution, we structured graphs as three-way data where the clustering is the principal objective. We propose an extension of LBM to tackle the co-clustering of multiple undirected/directed graphs where each cell of the diagonal is not necessarily equal to an even number as conventionally considered in community detection. To do this, we adopt an EM-type approach to refer to the Expectation-Maximization algorithm [Dempster et al., 1977, McLachlan and Peel, 2000]) and not Classification EM [Celeux and Govaert, 1992]. Furthermore, we will show that this purpose can be viewed as an implicit consensus clustering from Multiple Graphs.

[1] In the paper, to distinguish between a model and its derived algorithm we use *typewriter font* for an algorithm, thereby DC-SBM is the model and `DC-SBM` its derived algorithm.

PSBM DC-SBM/Croinfo

FIGURE 3.3: Political blogs dataset: Clustering with `PSBM` and `DC-SBM/Croinfo`.

3.3.2 Poisson Latent and Stochastic Block Models

As we mentioned earlier, Poisson SBM, even DC-SBM, are particular cases of Poisson LBM insofar as the latter can model matrices, symmetric or not, oriented or non-oriented graphs, numbers of row clusters and columns clusters not necessarily equal ($g \neq m$) and finally with proportions of clusters equal or not. Therefore the transition from LBM to SBM is easy to show. Thereby, for undirected graph, the maximization of (2.8) leads to maximizing

$$L_C(\widetilde{\mathbf{Z}}, \boldsymbol{\Omega}) + 2H(\widetilde{\mathbf{Z}}),$$

which is proportional to

$$\sum_{i,k} \tilde{z}_{ik} \log \pi_k + \frac{1}{2} \sum_{i \neq j, k \neq \ell} \tilde{z}_{ik} \tilde{w}_{j\ell} \log \mathcal{P}(x_{ij}; x_{i.} x_{.j} \gamma_{k\ell})$$
$$+ \frac{1}{2} \sum_{i,k} \tilde{z}_{ik} \log \mathcal{P}(x_{ii}; x_{i.} x_{i.} \gamma_{kk}) - \sum_{i,k} \tilde{z}_{ik} \log \tilde{z}_{ik}.$$

The main differences between them are a) considering the Poisson SBM, the last term, which concerns the diagonal of X, is skipped and it does not take into account the degree of nodes, unlike LBM which considers the diagonal elements. b) with Poisson LBM, $x_{ij}|z_{ik}w_{j\ell} = 1 \sim \mathcal{P}(x_{i.}x_{.j}\gamma_{k\ell})$, while with SBM $x_{ij}|z_{ik}w_{j\ell} = 1 \sim \mathcal{P}(\gamma_{k\ell})$. Notice that $\gamma_{k\ell}$ depends only on the block $k\ell$ and not on the margins. Thereby, starting from PLBM, next we will see how to take into account the sparsity often present in the graphs.

In Figure 3.4 we report the graphical models of Poisson models discussed in the chapter. To clarify expectations and the impact of this parameterization, On political blogs dataset[2], we applied the clustering algorithms derived from SBM, PLBM, and SPLBM from 30 random initializations and measure the accuracy. Figure 3.5 shows the interest of SPLBM, which takes into account the sparsity often present in a graph network.

The properties of this parameterization prompt us to adopt it for co-clustering. In fact, when $i = j$ we have $z_{ik} = w_{jk}$ and for $k = 1, \ldots, g$ we have $\pi_k = \rho_k$. Next, to avoid confusion between all the rows and columns that are identical in our case, we still keep the notations using the z_{ik}'s and $w_{j\ell}$'s.

[2] https://dl.acm.org/citation.cfm?id=1134277

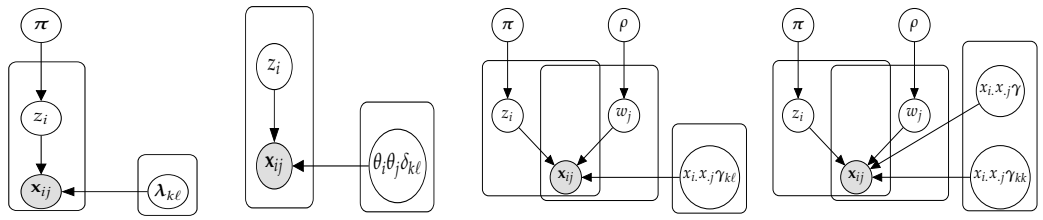


FIGURE 3.4: Graphical models: z_i is the label of row i, w_j is the label of column j.

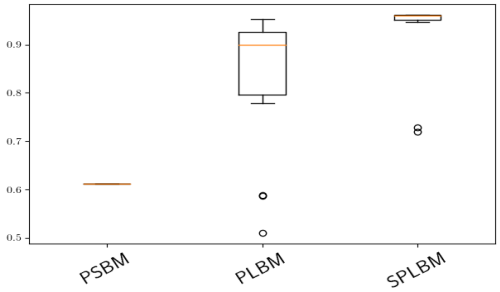


FIGURE 3.5: Political blogs dataset: Comparison of `PSBM`, `PLBM`, and `SPLBM` in terms of accuracy.

3.3.3 TSPLBM with multiple graphs

Our proposal Tensor SPLBM for multiple graphs, considers 3D data matrix $\mathcal{X} = [\mathbf{x}_{ij}] \in \mathbb{R}^{n\times n\times v}$ where n is the number of nodes, and v the number of graphs (slices). Figure 3.2 presents a tensor data with v graphs. Assuming the independence per graph, the conditional Poisson pdf is given by

$$\prod_{i,j=1}^{n}\prod_{k=1}^{g}\prod_{b=1}^{v}\{\mathcal{P}(x_{ij};x_{i.}^{b}x_{.j}^{b}\gamma_{k\ell}^{b})\}^{z_{ik}w_{j\ell}}.$$

As $\mathcal{X}$ is symmetric per slice b, when $i=j$ we have $z_{ik}=w_{jk}$ and for $k=1,\ldots,g$ we have $\pi_k=\rho_k$, and we have to optimize $\frac{1}{2}\mathcal{L}_C(\widetilde{\mathbf{Z}},\widetilde{\mathbf{W}},\mathbf{\Omega})+H(\widetilde{\mathbf{Z}})$ which takes the following form

$$\sum_{i,k}\tilde{z}_{ik}\log\pi_k+\frac{1}{2}\left(\sum_{i,j,k}\tilde{z}_{ik}\tilde{w}_{jk}\sum_{b=1}^{v}\log\mathcal{P}(x_{ij}^{b};x_{i.}^{b}x_{.j}^{b}\gamma_{kk}^{b})+\sum_{i\neq j,k\neq\ell}\tilde{z}_{ik}\tilde{w}_{j\ell}\sum_{b=1}^{v}\log\mathcal{P}(x_{ij}^{b};x_{i.}^{b}x_{.j}^{b}\gamma^{b})\right)+H(\widetilde{\mathbf{Z}}). \tag{3.2}$$

After some algebraic calculations and simplifications, and considering that $x_{k.}^{b}=\sum_i\tilde{z}_{ik}x_{i.}^{b}=\sum_j\tilde{w}_{jk}x_{.j}^{b}=x_{.k}^{b}$, $x_{kk}^{b}=\sum_{i,j}\tilde{z}_{ik}\tilde{w}_{jk}x_{ij}^{b}$, and $N_b=\sum_{i,j}x_{ij}^{b}$, this leads (up a constant) to :

$$\sum_{i,k} \tilde{z}_{ik} \log \pi_k + \frac{1}{2} \sum_b \sum_k (x^b_{kk} \log(\gamma^b_{kk}) - x^b_{k.} x^b_{.k} \gamma^b_{kk})$$
$$+ \frac{1}{2} \sum_b \left((N_b - \sum_k x^b_{kk}) \log(\gamma^b) - (N_b^2 - \sum_k x^b_{k.} x^b_{.k}) \gamma^b \right) + H(\widetilde{\mathbf{Z}})$$
$$= \sum_{i,k} \tilde{z}_{ik} \log \pi_k + H(\widetilde{\mathbf{Z}})$$
$$+ \frac{1}{2} \sum_b \left(\sum_k \left[x^b_{kk} \log(\frac{\gamma^b_{kk}}{\gamma^b}) - x^b_{k.} x^b_{.k} (\gamma^b_{kk} - \gamma^b) \right] + N_b (\log(\gamma^b) - N_b^2 \gamma^b) \right).$$

3.3.4 Variational Inference

To estimate the parameters of the model, we rely on the Variational EM algorithm [Govaert and Nadif, 2005], and we extend it to multiple graphs. In the sequel, the proposed algorithm is referred to as `TSPLBM`.

E-step. It consists in computing, for all i, j, k the posterior probabilities $\tilde{z}_{ik}$ and $\tilde{w}_{jk}$ given the estimated parameters $\boldsymbol{\Omega}$. As $\sum_k \tilde{z}_{ik} = \sum_k \tilde{w}_{jk} = 1$, using the corresponding Lagrangians, up to terms which are not function of $\tilde{z}_{ik}$, leads to (See Appendix **??**)

$$\tilde{z}^{(t+1)}_{ik} \propto \log \pi_k + \frac{1}{2} \left(\sum_{j,k} \tilde{z}^{(t)}_{jk} \sum_{b=1}^{v} \mathcal{P}^{ijb}_{kk} + \sum_{j \neq i, k \neq \ell} \tilde{z}^{(t)}_{j\ell} \sum_{b=1}^{v} \mathcal{P}^{ijb}_{k\ell} \right), \tag{3.3}$$

where $\mathcal{P}^{ijb}_{kk} = \log \mathcal{P}(x^b_{ij}; x^b_{i.} x^b_{.j} \gamma^b_{kk})$ and with $k \neq \ell$, $\mathcal{P}^{ijb}_{k\ell} = \log \mathcal{P}(x^b_{ij}; x^b_{i.} x^b_{.j} \gamma^b)$. The update of $\tilde{z}^{(t+1)}_{ik}$ is described in Appendix, and $\tilde{z}^{(t)}_{ik}$ represents the value of $\tilde{z}_{ik}$ in the previous iteration (t).

M-step. Given the previously computed posterior probabilities $\widetilde{\mathbf{Z}}$, the M-step consists in updating, $\forall k$, the parameters π_k, γ^b_{kk} and γ^b. The estimated parameters are defined as follows. First, taking into account the constraints $\sum_k \pi_k = 1$, it is easy to show that $\pi_k = \frac{\sum_i \tilde{z}_{ik}}{n}$. Secondly, it is easy to obtain for all b, k (See Appendix C)

$$\gamma^b_{kk} = \frac{\sum_{i,j} \tilde{z}_{ik} \tilde{z}_{jk} x^b_{ij}}{\sum_i \tilde{z}_{ik} x^b_{i.} \sum_j \tilde{z}_{jk} x^b_{.j}} = \frac{x^b_{kk}}{[x^b_{k.}]^2} \text{ and,}$$

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

The `TSPLBM` algorithm (Algorithm 7) for multiple graphs (MG), alternates the two previously described steps Expectation-Maximization. At the convergence, a hard co-clustering is deduced from $\tilde{z}_{ik}$'s using the maximum a posteriori principle.

Algorithm 7: TSPLBM

Input: $\mathcal{X}$, g.
Initialization: $\mathbf{Z}^{(0)}$ randomly and compute $\mathbf{\Omega}^{(0)}$, $t = 0$
repeat
  E-Step: Compute $\tilde{z}_{ik}^{(t+1)}$
  $\tilde{z}_{ik}^{(t+1)} \propto \pi_k \exp\left(\sum_j \tilde{z}_{jk}^{(t)} \sum_{b=1}^{v} x_{ij}^b \log(\frac{\gamma_{kk}^b}{\gamma^b})\right)$
  M-Step: Update $\mathbf{\Omega}^{(t+1)} = (\pi_k^{(t+1)}, (\gamma_{kk}^b)^{(t+1)}, (\gamma^b)^{(t+1)})$ given by
  $\pi_k = \frac{\sum_i \tilde{z}_{ik}}{n}$, $\gamma_{kk}^b = \frac{x_{kk}^b}{[x_{k.}^b]^2}$, and $\gamma^b = \frac{N_b - \sum_k x_{kk}^b}{N_b^2 - \sum_k [x_{k.}^b]^2}$
until *convergence*;
return Z, Ω

3.4 Experiments

In our experiments, we aim to discuss three important questions about (i) The importance of considering multiple graphs simultaneously on clustering results through TSPLBM and comparison with baselines considering one graph each time. (ii) The second point shows how the proposed model can help with the interpretation of the obtained results. (iii) And finally, we made a parallel between the proposed approach and clustering ensemble, and we compare implicit consensus obtained by TSPLBM and the explicit consensus achieved by the clustering ensemble method.

3.4.1 Datasets and evaluation

We use four datasets with a different number of graphs (slices) and clusters. Table 3.1 shows the characteristics of datasets.

TABLE 3.1: Characteristics of datasets.

Datasets	Type	#Graphs	#Node	#Cluster
DBLP1	Text	3	2223	3
DBLP3	Text	3	12550	10
Nus-Wide-8	Text+Images	6	2738	8
Amazon-products-10	Text+Images	7	9897	10

DBLP1 and DBLP3: The two datasets DBLP1 and DBLP3 are document datasets constructed from the global DBLP[3] dataset. The clusters are represented by journals/conferences where the papers are published. We selected three journals ((and conferences) for DBLP1, namely Discrete Applied Mathematics, IEEE software, and SIGIR. For DBLP3, we selected ten journals (and conferences), which are ICC, IJCAI', SIGMOD, Discrete Applied Mathematics, Electr. Notes Theor. Comput. Sci., DAC, GECCO, ICIP, ICCV, and Journal of

[3]https://aminer.org/citation

Systems and Software. We constructed three graphs. *Co-terms Title*, and *Co-terms Abstract*, are adjacency matrices representing the co-terms between documents on the title and abstract, respectively. The *Co-terms* $\mathcal{T}$ matrix is computed using $\mathcal{B}\mathcal{B}^\top$, where $\mathcal{B}$ is a binarized documents-terms matrix, then $\forall i, \mathcal{T}_{ii} > 0$. We also have *Co-authors* graph denoting the number of joint authors for two documents.

Nus-Wide-8 dataset: It is a part of the Nus-Wide images dataset[4] extracted using Flickr API. This dataset is composed of eight topics, namely Animals, Persons, Plants, Snow, Street, Temple, Town, and Wedding. We constructed six graphs — the *Co-tags* graph, which is an adjacency matrix of common tags between images. As described in the previous paragraph for *Co-terms* matrix, we used a binary matrix images-tags $\mathcal{M}$ to compute *Co-tags* matrix $\mathcal{H}$ by $\mathcal{M}\mathcal{M}^\top$. Other graphs are also created based on extracted features from images. The followed process to build graph similarity based on six extracted features form images including 64-D Color Histogram (CH), 144-D Color Correlogram (CORR), 73-D Edge direction histogram (EDH), 128-D Wavelet texture (WT), 225-D block-wise color moments (CW55). The computed similarity matrices are converted to adjacency matrices by putting one if the similarity is higher than ninety-seven percent quantile and zero otherwise.

Amazon-products-10 dataset: It is a part of the Amazon-products dataset[5], composed of product images. We consider ten product categories, namely Beauty, Digital music, Home and kitchen, Office products, Cell phones, Sports and outdoors, Health and personal care, Clothing-Shoes-Jewelry, Patio-garden, and Baby. We constructed seven graphs. The three first one *Similarity LBP*, *Similarity Haralick* and *Similarity Gabor* are constructed based on Low Rank Representation (LRR) method [Liu et al., 2013a] for three different features namely 256-D Local Binary Patterns (LBP), 216-D Haralick features [Haralick et al., 1973] (considering distances $d = 1 \ldots 9$, orientations θ = [0°, 45°, 90°, 135°]) and 192-D Gabor features [Chengjun Liu and Wechsler, 2001] (considering scales $\sigma = 1 \ldots 4$, orientations θ = [0°, 45°, 90°, 135°]). The computed similarity matrices are converted to adjacency matrices by putting one if the similarity is higher than ninety-seven percent quantile and zero otherwise. *Co-terms Title* and *Co-terms Description* are adjacency matrices representing the co-terms between the title and description of products, respectively. Finally, *Co-viewed* and *Co-purchased* are adjacency matrices $\mathcal{Y}$, where $\mathcal{Y}_{ij} = 1$ means that these two products are viewed (respectively purchased) simultaneously when users make a query.

Figure 3.6 shows all graphs (slices) for the Amazon-products-10 dataset. The dataset is composed of seven graphs. We notice that each slice has different structures and different degrees of complexity. Our `TSPLBM` input is a tensor (Node × Node × Graph) for each dataset DBLP1, DBLP3, Nus-Wide-8, and Amazon-products-10 with different sparsity 0.96, 0.99, 0.83, and 0.98 respectively.

[4] https://dl.acm.org/citation.cfm?id=1646452

[5] http://jmcauley.ucsd.edu/data/amazon/links.html

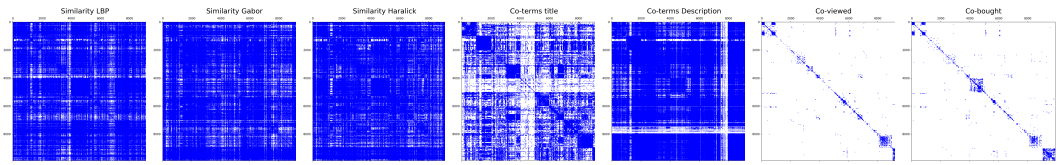


FIGURE 3.6: Amazon-products-10 dataset.

3.4.2 Algorithm evaluation

What is the impact of considering multiple graphs on clustering results?

We first compare TSPLBM applied on all graphs simultaneously with PSBM, PLBM, SPLBM used on each graph. The goal is to evaluate TSPLBM in terms of clustering with a comparison with the baselines. On the other hand, we aim to measure how the combination of different information through graphs, impacts, and improves results. Note that TSPLBM can be viewed as an ensemble method.

We perform 30 random initializations and compute Accuracy and Normalized Mutual Information (NMI) [Strehl and Ghosh, 2002] metrics by averaging all runs. The clustering accuracy noted (ACC) discovers the one-to-one relationship between two partitions and measures the extent to which each cluster contains data points from the corresponding class. However, NMI is based on Mutual Information (MI) and measures the amount of retrieved information considering our knowledge about the clusters and the obtained results by a clustering method while respecting the proportions of clusters.

In Figure 3.7, the performances of the four algorithms PSBM, PLBM, SPLBM, and TSPLBM on the four datasets, are reported. PSBM, PLBM, and SPLBM are applied on each slice (graph) separately. TSPLBM is applied to the tensor considering all graphs simultaneously.

We notice that, in most cases, TSPLBM is better than other algorithms applied to each graph and allows us to achieve the best trade-off. TSPLBM includes all graphs and also the graphs with a very complex structure. DBLP3 obtains the lowest results due to the complex structure of dataset composed of 12K papers with very close or complementary topics on computer science. We observe that PLBM and SPLBM do a better job than PSBM for all datasets on the more informative slices. It is also worth noting that PLBM does good performances in terms of Accuracy on DBLP1 and in terms of NMI on DBLP3. TSPLBM performs a natural consensus when considering all slices and allows us to obtain a unique partition at the end with good clustering results.

How can the proposed model help us in the interpretation of the obtained results?

The objective of this part is to analyze the obtained topics and demonstrate how the proposed model can help and then improve the interpretation of the obtained clusters.

The second analysis that we made is dimensionality reduction of topics-tags matrix using the correspondence analysis method (CA) [Benzecri, 1973, Nenadic and Greenacre, 2007]. The choice of CA is due to the connection between mutual information and chi-square, which is based in CA, see, e.g., [Govaert and Nadif, 2018]. The matrix topic-tags $\mathbf{Z}^T\mathcal{M}$ is constructed from *image-tags* $\mathcal{M}$ based on obtained topics (or partition) $\mathbf{Z}$ obtained by TSPLBM.

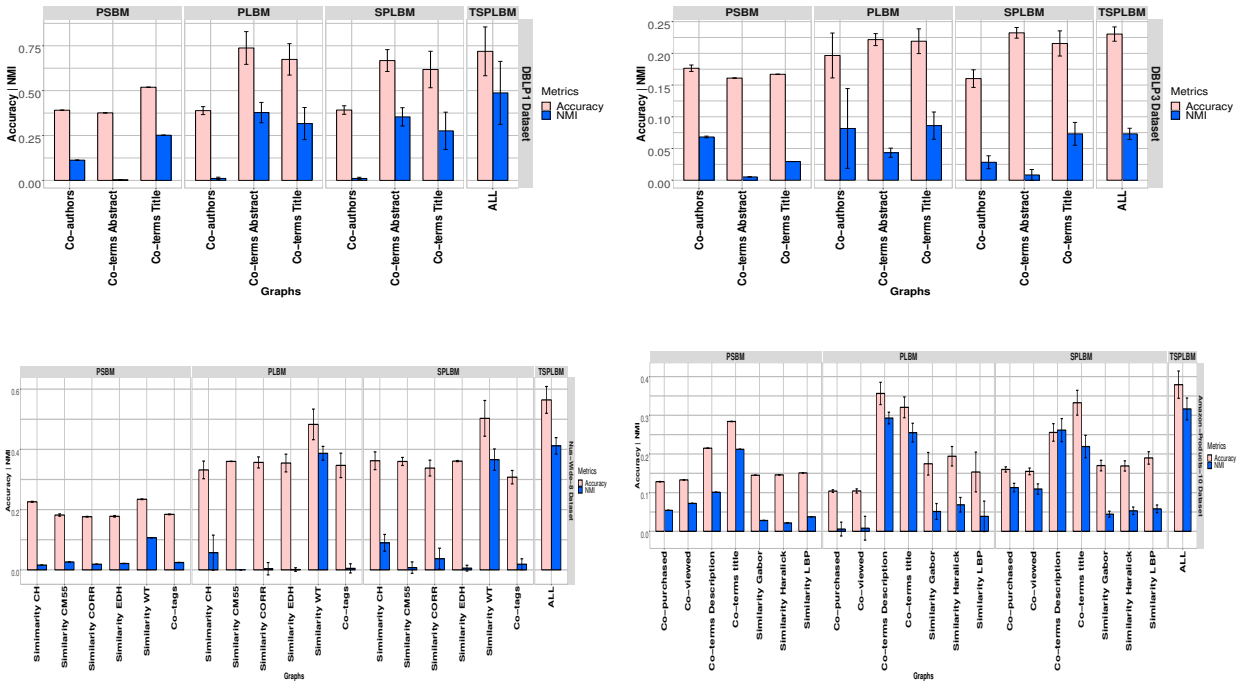


FIGURE 3.7: Comparison in terms of Accuracy and NMI for all datasets with PSBM, PLBM, SPLBM and TSPLBM.

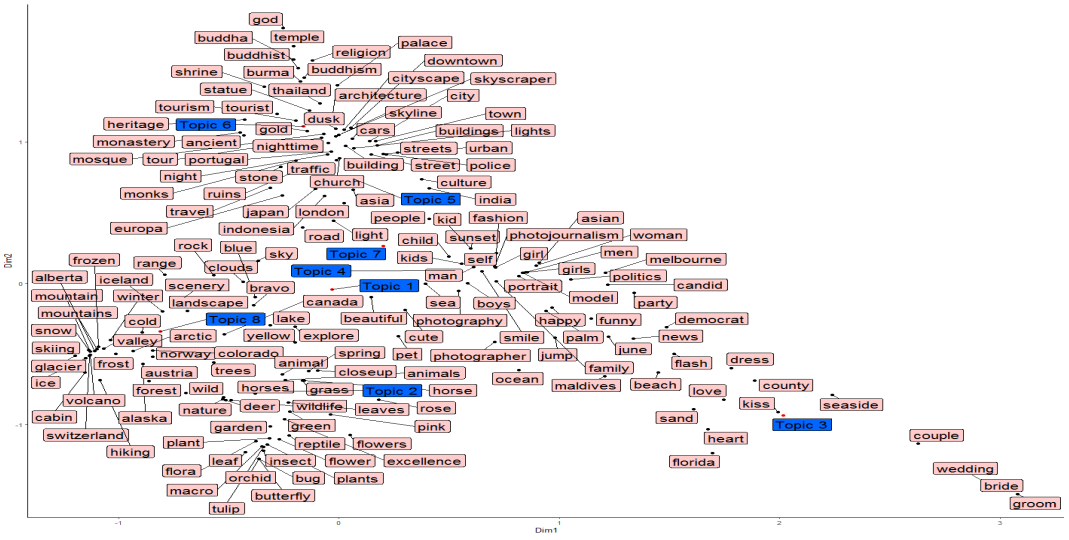


FIGURE 3.8: CA applied on topic-tags matrix.

In Figure 3.8, are projected the tags and topics on the two first dimensions of CA including the top tags in terms of contribution[6] on the CA results.

We can notice that there are some close topics and other very different one. For instance,

[6]With CA each tag contributes to the inertia of each axis. The contribution of a tag to axis α is expressed as a percent of the inertia for axis α.

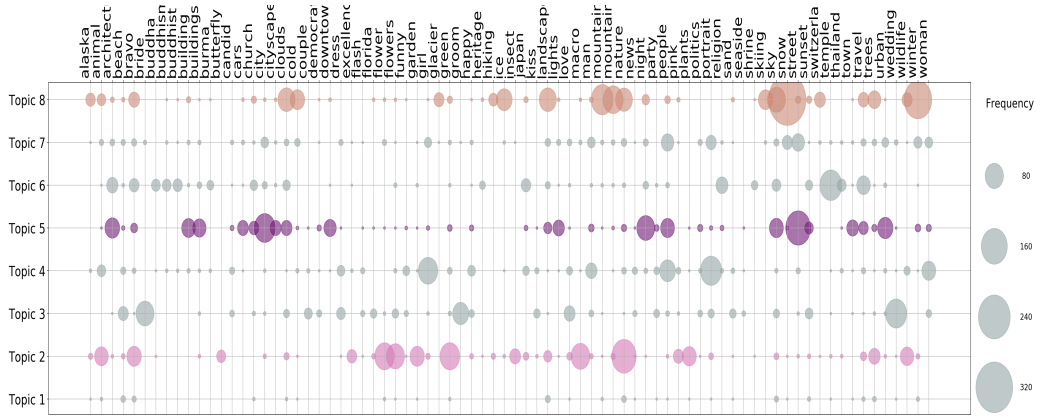


FIGURE 3.9: Topic-tags frequencies matrix using top CA contributed tags.

topic 3 about weddings is opposed to topics 8 and 6 about *snow* and *temple* considering the first and the second dimension respectively. On the other hand, we can see that topics 1 and 2 about plants and animals are close.

igure 3.9 presents the tags whose contribution is important. We show the frequencies of each term for each topic. For topics 2 and 5 (pink and purple color respectively), we can see that the four top tags are *Nature, Green, Macro, and Flower* related to Plants topic and *Street, City, Night and Architect* related to Town topic.

Based on the *Co-tags* graph and the obtained topics, we construct a graph of image clusters linked by edges representing the intensity of joint tags between all topics, this can be computed by $\mathbf{Z}^\top \mathcal{H} \mathbf{Z}$ where $\mathbf{Z}$ is obtained by `TSPLBM`, and $\mathcal{H}$ is the co-tags matrix (see figure 3.10). We can notice that there are some topics with a strong relationship like *plants-snow* and *town-persons*. On the other hand, some topics with a weak link like *animals-town* and *animals-temple*. This representation highlights that there are some tags used with confused meaning. In this context, it is possible to use tensor models for tags completion and tags correction [Tang et al., 2017, Veit et al., 2017].

3.4.3 Implicit consensus VS explicit consensus

In the first part of our experiments, we have observed that `TSPLBM` applied on all slices simultaneously is, in most cases, better than other algorithms. As we are in an unsupervised context, we have found it helpful to run the calculation with several different random initial conditions and take the best result in terms of maximum log-likelihood, overall runs.

Figure 3.11 shows the 30 performed runs sorted according to Normalized log-likelihood (NL), which is the objective function of `TSPLBM`. We also draw the ACC and NMI curve according to the 30 runs. We observe that for DBLP1, the best runs leading to maximal NL are the best runs in terms of clustering (ACC and NMI). However, this observation is not noticed in all datasets; for instance, some best runs can achieve less good results in terms of ACC and NMI. This problem is recurrent with all unsupervised methods where the best runs in terms of the objective function are not necessarily the best ones in terms of clustering. On the other hand, we may see the proposed model as an implicit consensus model for graphs

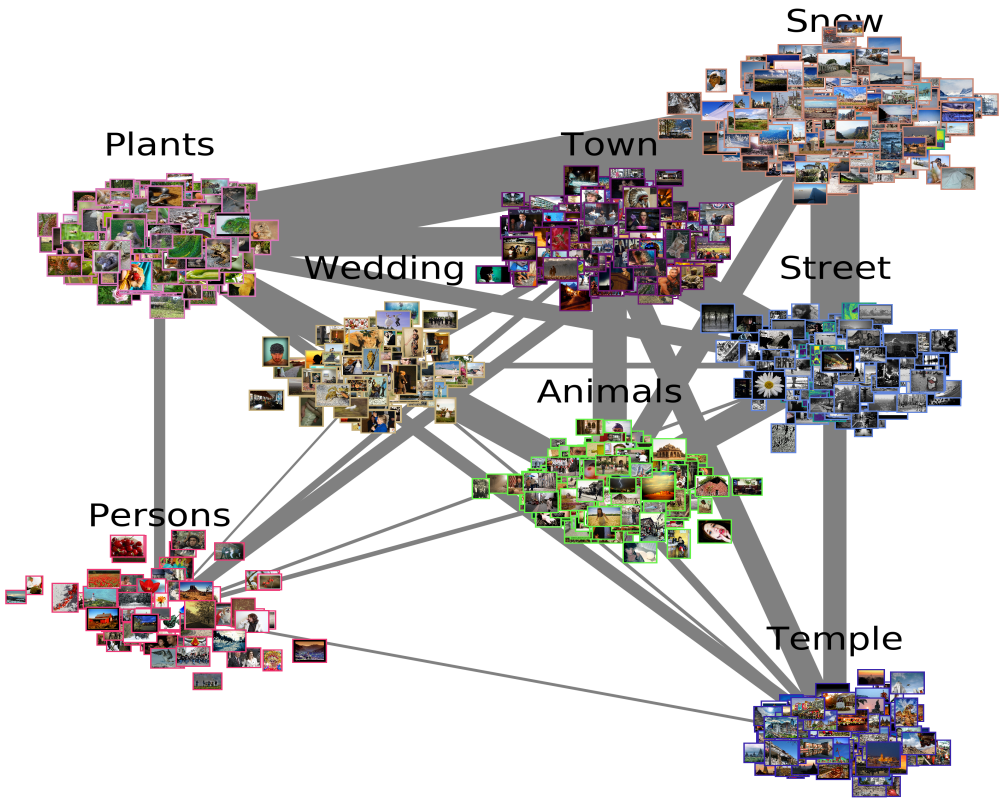


FIGURE 3.10: Co-tags graph of `Nus-Wide-8`.

clustering, and it is tempting to compare the proposed model to ensemble-based clustering methods.

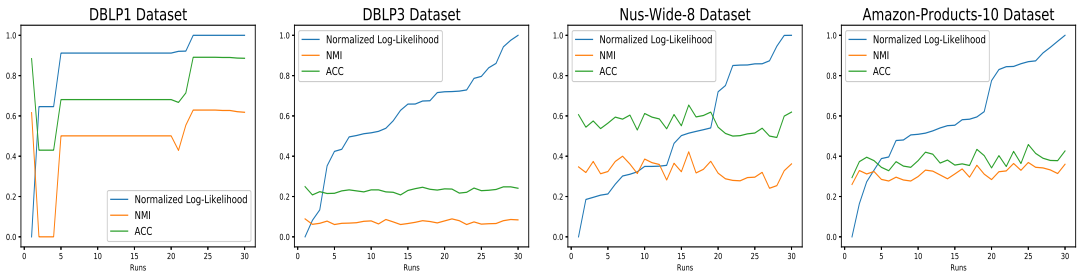


FIGURE 3.11: Normalized Log-likelihood *vs* NMI and ACC for all runs.

The first works about consensus or ensemble classification have emerged in the context of supervised learning; see for instance [Maclin and Opitz, 1997, Schapire, 2003, Dietterich, 2000]. However, only the majority voting type algorithms work on the model output level, and the most well-known classification ensembles approaches are based on different variants

of voting [Bauer and Kohavi, 1999, Crammer et al., 2007, Gao et al., 2013]. This approach has been extended to unsupervised learning [Strehl and Ghosh, 2002, Vega-Pons and Ruiz-Shulcloper, 2011]. A clustering ensemble, also known as a consensus clustering or clustering aggregation, is defined in the same manner as for classification [Hanczar and Nadif, 2012, Alqurashi and Wang, 2019, Yu et al., 2019]. It consists in combining multiple clustering models (partitions) into a single consolidated partition.

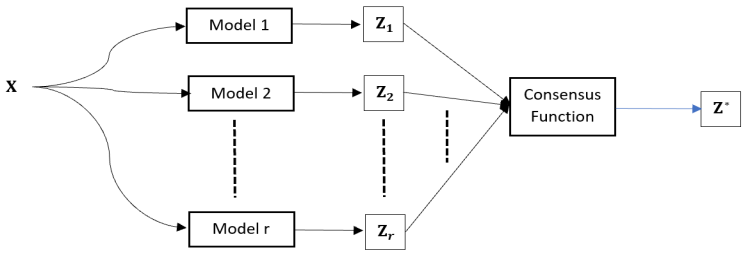


FIGURE 3.12: Consensus clustering.

In other words, from r partitions $\{\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3, \ldots, \mathbf{Z}_r\}$, a consensus clustering leads to a unique partition $\mathbf{Z}^*$. Based on consensus functions, many approaches exist; see for instance [Strehl and Ghosh, 2002, Hanczar and Nadif, 2012] (see figure 3.12).

In [Strehl and Ghosh, 2002], the authors introduced three ensemble clustering methods that can produce a consensus partition. All of them consider the consensus problem on a hypergraph representation of the set of partitions. More specifically, each partition is a binary classification matrix (with objects in rows and clusters in columns) where the concatenation of all the set defines the hypergraph. Figure 3.13 presents this matrix and different steps to construct a combination of these different graphs of clusters, emerged from different partitions, to obtain a unique graph. To this end, we rely on the three hypergraph clustering-based approaches proposed by Strehl and Ghosh [2002], namely `CSPA` (Cluster-based Similarity Partitioning Algorithm), `HGPA` (HyperGraph Partitioning Algorithm), and `MCLA` (Meta-CLustering Algorithm).

	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$
x_1	1	1	2
x_2	1	1	2
x_3	1	1	1
x_4	2	2	1
x_5	2	4	1
x_6	2	4	3
x_7	3	3	3
x_8	3	3	3

	$\mathbf{Z}_1$			$\mathbf{Z}_2$				$\mathbf{Z}_3$		
x_1	1	0	0	1	0	0	0	0	1	0
x_2	1	0	0	1	0	0	0	0	1	0
x_3	1	0	0	1	0	0	0	1	0	0
x_4	0	1	0	0	1	0	0	1	0	0
x_5	0	1	0	0	0	0	1	1	0	0
x_6	0	1	0	0	0	0	1	0	0	1
x_7	0	0	1	0	0	1	0	0	0	1
x_8	0	0	1	0	0	1	0	0	0	1

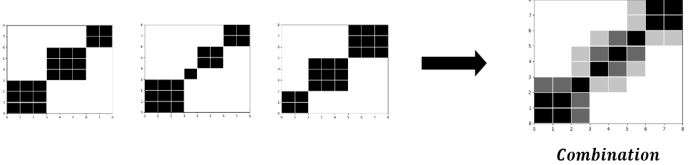


FIGURE 3.13: Graphs clustering similarity.

To improve clustering results of TSPLBM we will adopt the ensemble approach. We explore in the next part, how implicit consensus clustering through TSPLBM behaves compared to explicit consensus through cluster ensembles of multiple graphs. In Figure 3.14, we report the proposed approach to compare TSPLBM with the clustering ensemble methods proposed by Strehl and Ghosh [2002]. To do this, we used the implementation of python package Cluster_Ensembles[7]. It relies on CSPA, HGPA, and MCLA and returns the best results in terms of the mean of NMI between the obtained consensus clustering $\mathbf{Z}^*$ and the different clustering solutions $\{\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3, \ldots, \mathbf{Z}_r\}$. Therebey, with TSPLBM, we select the top ten runs maximizing log-likelihood then we carry out the consensus by using the cluster-ensembles methods. With SPLBM, PLBM, and PSBM, we consider two steps. The first step is the same as that used with TSPLBM to select the top ten runs and apply the cluster-ensembles methods. The second one consists in applying another clustering consensus between graphs to obtain a unique partition.

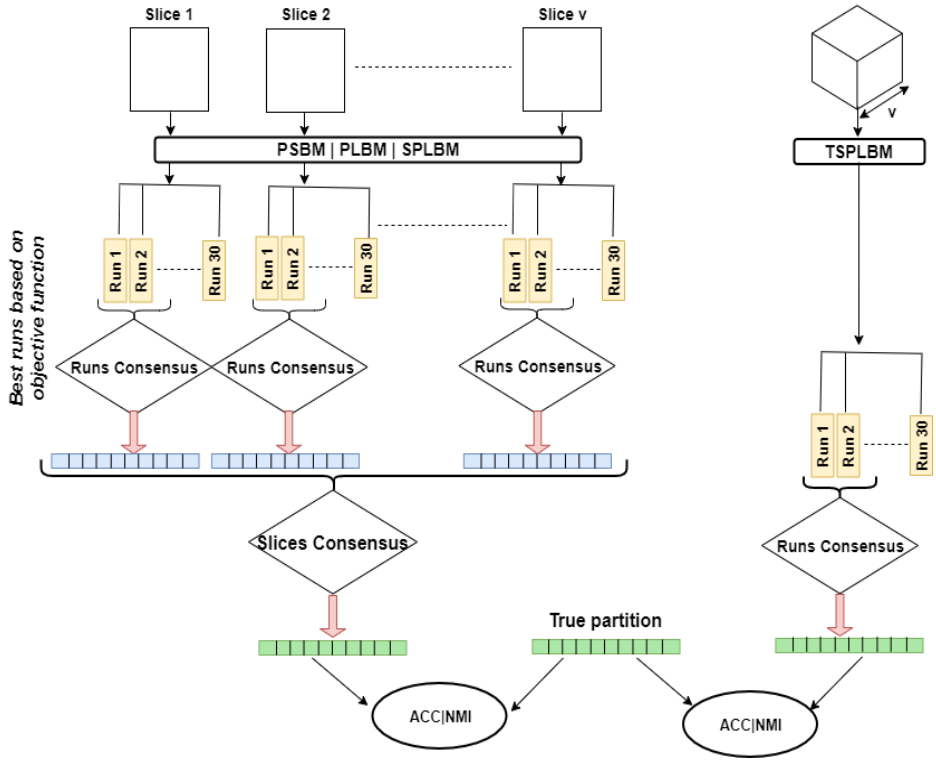


FIGURE 3.14: Comparison approach.

In Figure 3.15 are reported the obtained results in terms of NMI using the comparison approach described above. We can notice that TSPLBM achieves the highest NMI for all datasets. SPLBM does a better or similar job than PLBM on three datasets. Unlike PSBM, which obtains the lowest NMI measures on all datasets. Our approach provides good results and can be used to obtain the most appropriate partition when dealing with multiple graphs.

[7] https://pypi.org/project/Cluster_Ensembles/

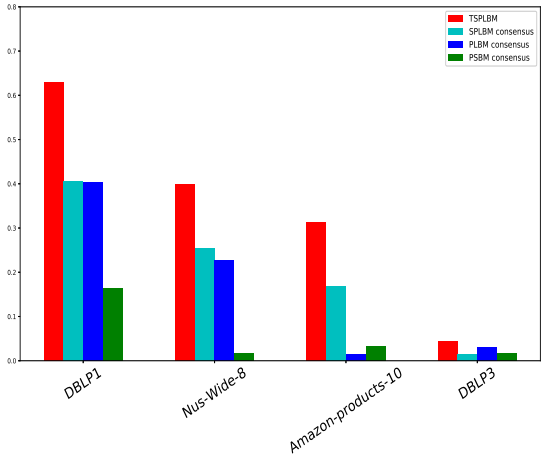


FIGURE 3.15: Consensus based NMI comparison.

3.5 Conclusion

It is well known that the traditional Poisson SBM fails to detect relevant clusters of edges, this requires a degree-corrected SBM (DC-SBM). Drawing on this, we first established some connections between Poisson SBM and the corrected version DC-SBM with Poisson LBM commonly used for the co-clustering of contingency tables. We justified the extension of the latter to deal with multiple graphs clustering. To take into account the sparsity of the tensor, we modified the parametrization of the model and proposed a Tensor SPLBM (TSPLBM). We derived, thereby, an EM-like learning algorithm called `TSPLBM` capable of performing clustering from a tensor data. On real datasets of text and image graphs, we have shown that `TSPLBM`, is better than the cited baselines algorithms in terms of clustering.

On the other hand, we can note that the proposed `TSPLBM` algorithm can be seen as an implicit consensus clustering between multiple graphs. To reinforce our idea that `TSPLBM` can be used in this sense, a comparative study with explicit consensus through ensemble clustering methods was realized. Experiments on several real graphs datasets highlight the effectiveness of `TSPLBM`. Thereby, this work gives an extra dimension to LBM as an ensemble method.

Finally, we have seen that our approach has made it possible to propose a like-EM learning algorithm. Thus, we can easily develop a like-Classification EM version. To do this, all that is needed is to insert a classification step between E and M steps. This could lead to propose an extension of `DC-SBM` for multiple graphs.

Chapter 4

Latent Block Regression Model

4.1 Introduction

In previous chapters, we have seen the role of unsupervised learning through model-based co-clustering. In the present chapter, we extend the interest of model-based approaches to supervised learning by combining a co-clustering and regression model in a unified framework. This is the objective of the *cluster-wise* model, which aims to discover clusters and fit a linear model per cluster.

The *cluster-wise* linear regression algorithm CLR (or Latent Regression Model) is a finite mixture of regressions and one of the most commonly used methods for simultaneous learning and clustering [Späth, 1979, De Sarbo and Corn, 1988]. It aims to find clusters of entities such as the overall sum of squared errors from regressions performed over these clusters is minimized. Specifically, $\mathbf{X} = [x_{ij}] \in \mathbb{R}^{n \times d}$ is the covariate matrix and $\mathbf{Y} \in \mathbb{R}^{n \times 1}$ the response vector, the *cluster-wise* method aims to find g clusters $C_1, \ldots, C_g$ and regression coefficients $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^{d \times 1}$ by minimizing the following objective function:

$$\sum_{k=1}^{g} \sum_{i \in C_k} (y_i - \sum_{j=1}^{d} \beta_j^{(k)} x_{ij} + b_k)^2 \quad \text{where,}$$

- y_i is the value of the dependent variable for subject/observation i defined by $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})$,
- x_{ij} is the value of the j-th independent variable for subject/observation i,
- $\beta_j^{(k)}$ is the j-th coefficient of multiple regression and b_k is the *intercept*.

Various adjustments have been made to this model to improve its performance in terms of clustering and prediction. In our contribution, we propose to embed the co-clustering and regression in the model.

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 high dimensional data sparse or not, co-clustering turns out to be more beneficial than one-sided clustering [Ailem et al., 2017, Ailem et al., 2017, Salah and Nadif, 2017], even if one is interested in clustering along one dimension only (see section 3.2.1). Thereby, co-clustering is the guiding principle of this chapter.

Although co-clustering has become popular in unsupervised learning, few works are devoted to its embedding in supervised learning. We can mention [Deodhar and Ghosh, 2010],

where the authors proposed the SCOAL approach (Simultaneous Co-clustering and Learning model) leading to co-clustering and prediction for binary data; they generalized the model to continuous data. However, this model does not take into account the sparsity of data, in the sense that it does not lead to homogeneous blocks. The obtained results in terms of *Mean Square Error* (MSE) are good, but in terms of co-clustering (homogeneity of co-clusters), no analysis has been presented. This model is also related to the soft PDLF (Predictive Discrete Latent Factor) model [Agarwal and Merugu, 2007], where the value of response y_{ij}'s in each co-cluster is modeled as a sum of $\beta^T x_{ij} + \delta_{k\ell}$ where β is a global regression model while $\delta_{k\ell}$ is a co-cluster specific offset. More recently, in [Vu and Aitkin, 2015] the authors proposed an algorithm taking into account only row covariates information to realize co-clustering and regression simultaneously. To this end, the authors are based on the latent block models [Govaert and Nadif, 2008]. In our contribution, we propose to rely also on this model but by considering both row and column covariates.

The proposed Latent Block Regression Model (LBRM) is an extension of finite mixtures of regression models where the co-clustering is embedded. It allows us to deal with co-clustering and regression simultaneously while taking into account covariates. To estimate the parameters we rely on a *Variational* Expectation-Maximization algorithm [Govaert and Nadif, 2005] referred to as VEM-LBRM. Figure 4.1 presents an illustration of the VEM-LBRM goal. Taking the recommendation problem as an example, we start from historical data of users' evaluation of items represented by rating matrix, combined with users and items features, for example, u_i and m_j, respectively. The VEM-LBRM algorithm deals with the co-clustering of users and items simultaneously while leading regression models per block. Figure 4.1 provides an overview of the expected results. Furthermore, the proposed model can be used for other application like microarray analysis or predicting unknown or missing data values.

The remainder of this chapter is organized as follows. Section 4.2 presents a brief description of recommender systems types. Section 4.3 presents the LBRM model from a statistical point of view through a graphical model. Section 4.4 details the proposed VEM-LBRM algorithm. Section 4.5 is devoted to experimental results on synthetic and real-world data sets; also evaluation of VEM-LBRM and comparison with competitive methods are reported. Section 4.6 concludes this chapter and provides some directions for future work.

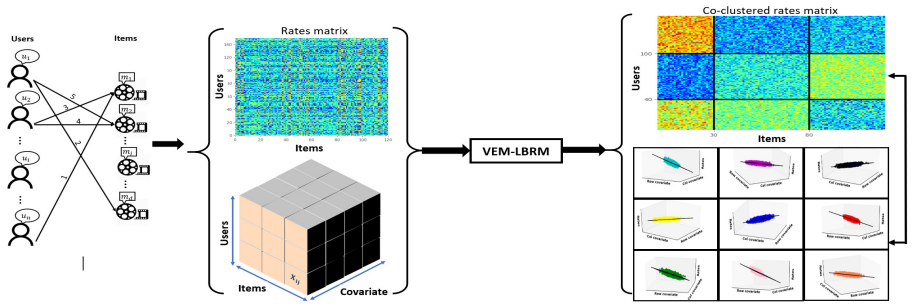


FIGURE 4.1: General VEM-LBRM algorithm operation.

4.2 Recommendation systems

Recommender systems (RSs) have evolved considerably in recent years. They are used in different fields of application, such as the recommendation of films, books, music, information, and various products. RSs are tools that predict the preferred product (or item) to a user (or customer). The term "Item" is generally used to refer to what we want to recommend to customers. Several recommendation techniques have been developed, to predict the most appropriate items for users (or clients) by addressing the problem of recommendation in different ways [2,13]. The most popular recommendation systems are (see figure 4.2):

- Content-based: It allows us to recommend items that are similar to the ones that the user liked in the past. The similarity between items is computed based on their characteristics and using different similarity measures such as *cosine similarity* leading to various type of content based recommender systems approaches.
- Collaborative filtering: This is the best-known type of recommendation system. This intuitive and straightforward approach allows us to recommend items that other users, with similar profiles and tastes, liked. The similarity between two users is calculated based on their historical ratings of products. This is why, Collaborative filtering is called "people-to-people correlation."

However, there are few works tackled the problem of hybrid recommendation systems. Hybrid recommendation systems combine *content-based* and *collaborative filtering* approaches. They help in addressing the sparsity and cold start issues as well as improve the results of recommendations. The advantage of hybrid approaches consists of using simultaneously, the available information about items, and the history of user-related interactions. In this chapter, we develop a hybrid recommender system through a suitable co-clustering algorithm. This leads to highlight groups of users (through Clustering) having similar profiles, but also to decide if a new item is of interest to the user (through a regression model).

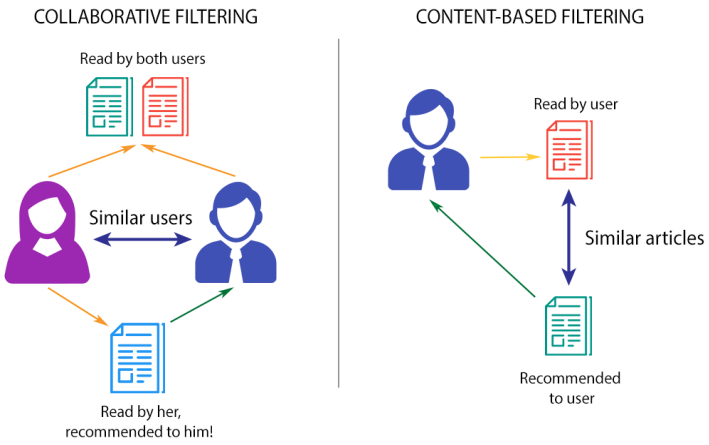


FIGURE 4.2: Recommendation techniques.

4.3 From Clusterwise regression to co-clusterwise regression

4.3.1 Co-clustering and LBM

Given an $n \times d$ data matrix $\mathbf{X} = (x_{ij}, i \in I = \{1,\ldots,n\}; j \in J = \{1,\ldots,d\})$, and $\boldsymbol{\Omega} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\lambda})$, the parameter of LBM 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 $\boldsymbol{\lambda} = (\lambda_{k\ell}; k = 1,\ldots g, \ell = 1,\ldots,m)$ where $\lambda_{k\ell}$ is the parameter of the distribution of block $k\ell$. The complete data log-likelihood of LBM leads to $L_C(\mathbf{X},\mathbf{Z},\mathbf{W},\boldsymbol{\Omega})$ which can be written as follows (see section 2.2.1)

$$\sum_{k=1}^{g} z_k \log \pi_k + \sum_{\ell=1}^{m} w_\ell \log \rho_\ell + \sum_{i=1}^{n}\sum_{j=1}^{d}\sum_{k=1}^{g}\sum_{\ell=1}^{m} z_{ik} w_{j\ell} \log \Phi_{k\ell}(x_{ij};\lambda_{k\ell}).$$

Note that the complete-data log-likelihood breaks into three terms: the first one depends on proportions of row clusters, the second on proportions of column clusters and the third on the pdf of each block or co-cluster. The objective is then to maximize the function $L_C(\mathbf{Z},\mathbf{W},\boldsymbol{\Omega})$.

For co-clustering of continuous data, the Gaussian latent block model can be used. For instance, note that it is easy to show that the minimization of the well-known criterion of

$$||\mathbf{X} - \mathbf{Z}\boldsymbol{\mu}\mathbf{W}^T||^2 = \sum_{k=1}^{g}\sum_{\ell=1}^{m}\sum_{i|z_{ik}=1}\sum_{j|w_{j\ell}=1}(x_{ij}-\mu_{k\ell})^2,$$

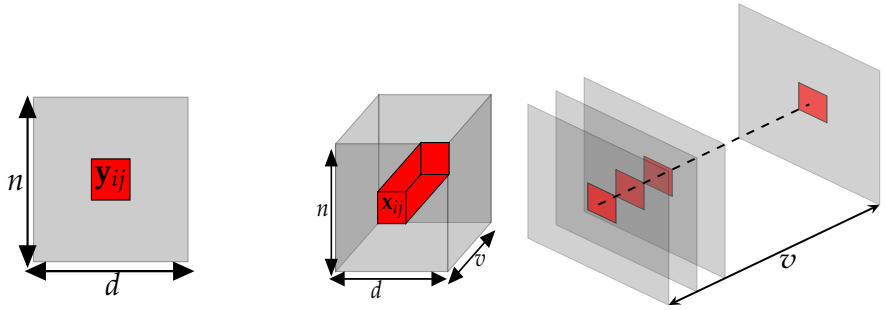
where $\mathbf{Z} \in \{0,1\}^{n\times g}$, $\mathbf{W} \in \{0,1\}^{d\times m}$ and $\boldsymbol{\mu} \in \mathbb{R}^{g\times m}$ is associated to Latent block Gaussian model whith $\lambda_{k\ell} = (\mu_{k\ell}, \sigma^2_{k\ell})$, the proportions of row clusters and column clusters are equal and in addition the variances of blocks are identical [Govaert and Nadif, 2013]. Note that 1) the characteristic of the latent block model is that the rows and the columns are treated symmetrically 2) the estimation of the parameters requires a variational approximation [Govaert and Nadif, 2005, Ghosh, 2009, Vu and Aitkin, 2015]. In the sequel, we will see how can we integrate a regression on co-clustering model.

4.3.2 Latent Block Regression Model (LBRM)

Hereafter, we propose a novel Latent Block Regression Model(LBRM) for co-clustering and learning simultaneously. The model considers the response matrix $\mathbf{Y} = [y_{ij}] \in \mathbb{R}^{n\times d}$ and the tensor covariate data $\mathcal{X} = [1, \mathbf{x}_{ij}] \in \mathbb{R}^{n\times d\times v}$ where n is the number of rows, d the number of columns, and v the number of covariates. Figure 4.3 presents data structure for the proposed LBRM.

In the following we propose the integration of mixture of regression [De Sarbo and Corn, 1988] per block in the Latent Block model (LBM) considering the distribution $\Phi(y_{ij}|\mathbf{x}_{ij};\lambda_{k\ell})$. We assume in the following the normality of Φ.

$$\Phi(y_{ij}|\mathbf{x}_{ij};\lambda_{k\ell}) = p(y_{i,j}|\mathbf{x}_{ij},\boldsymbol{\beta}_{k\ell},\sigma_{k\ell}) = \frac{1}{\sqrt{2\pi\sigma^2_{k\ell}}}\exp\left\{-\frac{1}{2\sigma^2_{k\ell}}(y_{ij}-\boldsymbol{\beta}^\top_{k\ell}\mathbf{x}_{ij})^2\right\}.$$



(A) Response matrix Y. (B) Tensor covariate data $\mathcal{X}$.

FIGURE 4.3: Data representation for proposed model.

With LBRM, the parameter $\boldsymbol{\Omega}$ is composed of row and column proportions $\boldsymbol{\pi}$, $\boldsymbol{\rho}$ respectively, the coefficents of regresssion $\boldsymbol{\beta} = \{\boldsymbol{\beta}_{11}, \ldots, \beta_{gm}\}$ with $\boldsymbol{\beta}_{k\ell}^{\top} = (\beta_{k\ell}^0, \beta_{k\ell}^1, \ldots, \beta_{k\ell}^v)$ where $\beta_{k\ell}^0$ represents the intercept of regression and $\boldsymbol{\sigma} = \{\sigma_{11}, \ldots, \sigma_{gm}\}$. The classification log-likelihood can be written :

$$L_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega}) = \sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell + \sum_{i,j,k,\ell} \log(\Phi(y_{ij}|\mathbf{x}_{ij}; \lambda_{k\ell})). \tag{4.1}$$

After some simplification, we obtain:

$$\begin{aligned} L_C(\mathbf{Z}, \mathbf{W}, \boldsymbol{\Omega}) &= \sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell \\ &- \frac{1}{2} \sum_{k,\ell} z_{.k} w_{.\ell} \log(\sigma_{k\ell}^2) - \frac{1}{2\sigma_{k\ell}^2} \sum_{i,j,k,\ell} z_{ik} w_{j\ell} (y_{ij} - \boldsymbol{\beta}_{k\ell}^{\top} \mathbf{x}_{ij})^2, \end{aligned}$$

with $z_{.k} = \sum_i z_{ik}$ et $w_{.\ell} = \sum_j w_{j\ell}$.

The graphical LBM and LBRM are presented in Figure 4.4. In LBRM, we deal with tensor data $\mathcal{X}$ and response matrix $\mathbf{Y}$ to achieve co-clustering and regression simultaneously.

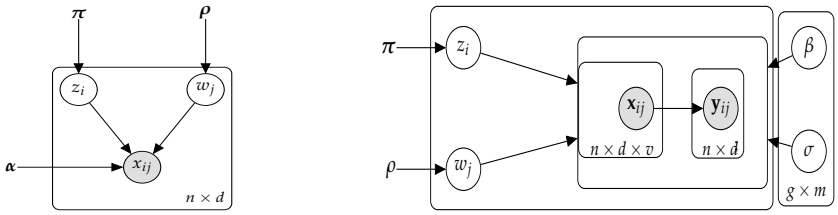


FIGURE 4.4: Graphical models: left mixture model without regression model(LBM), right proposed model(LBRM).

4.4 Variational EM algorithm

To estimate $\mathbf{\Omega}$, the EM algorithm [Dempster et al., 1977] is a candidate for this task. It maximizes the log-likelihood $f(\mathcal{X}, \mathbf{\Omega})$ w.r. to $\mathbf{\Omega}$ iteratively by maximizing the conditional expectation of the complete data log-likelihood $L_C(\mathbf{Z}, \mathbf{W}; \mathbf{\Omega})$ w.r. to $\mathbf{\Omega}$, given a previous current estimate $\mathbf{\Omega}^{(c)}$ and the observed data $\mathcal{X}$. Unfortunately, difficulties arise owing to the dependence structure among the variables x_{ij} of the model. To solve this problem an approximation using the [Neal and Hinton, 1998] interpretation of the EM algorithm can be proposed; see, e.g., [Govaert and Nadif, 2005, Govaert and Nadif, 2008]. Hence, the aim is to maximize the following lower bound of the log-likelihood criterion:

$$F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}; \mathbf{\Omega}) = L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega}) + H(\tilde{\mathbf{Z}}) + H(\tilde{\mathbf{W}}), \tag{4.2}$$

where $H(\tilde{\mathbf{Z}}) = -\sum_{i,k} \tilde{z}_{ik} \log \tilde{z}_{ik}$ with $\tilde{z}_{ik} = P(z_{ik} = 1|\mathcal{X})$, $H(\tilde{\mathbf{W}}) = -\sum_{j,\ell} \tilde{w}_{j\ell} \log \tilde{w}_{j\ell}$ with $\tilde{w}_{j\ell} = P(w_{j\ell} = 1|\mathcal{X})$, and $L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}; \tilde{\mathbf{\Omega}})$ is the fuzzy complete data log-likelihood (up to a constant). $L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}; \mathbf{\Omega})$ is given by

$$\begin{aligned} L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega}) &= \sum_{i,k} \tilde{z}_{ik} \log \pi_k + \sum_{j,\ell} \tilde{w}_{j\ell} \log \rho_\ell \\ &- \frac{1}{2} \sum_{k,\ell} \tilde{z}_{.k} \tilde{w}_{.\ell} \log(\sigma_{k\ell}^2) - \frac{1}{2\sigma_{k\ell}^2} \sum_{i,j,k,\ell} \tilde{z}_{ik} \tilde{w}_{j\ell} (y_{ij} - \boldsymbol{\beta}_{k\ell}^\top \mathbf{x}_{ij})^2. \end{aligned}$$

The maximization of $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega})$ can be reached by realizing the three following optimization: update $\tilde{\mathbf{Z}}$ by $\arg\max_{\tilde{\mathbf{Z}}} F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega})$, update $\tilde{\mathbf{W}}$ by $\arg\max_{\tilde{\mathbf{W}}} F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega})$ and update $\mathbf{\Omega}$ by $\arg\max_{\mathbf{\Omega}} F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega})$. In what follows, we detail the Expectation (E) and Maximization (M) step of the Variational EM algorithm for tensor data.

E-step. The E-step consists in computing, for all i, k, j, ℓ the posterior probabilities $\tilde{z}_{ik}$ and $\tilde{w}_{j\ell}$ maximizing $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega})$ given the estimated parameters $\mathbf{\Omega}_{k\ell}$. It is easy to show that, the posterior probability $\tilde{z}_{ik}$ maximizing $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega})$ (See Appendix A) is given by:

$$\tilde{z}_{ik} \propto \pi_k \exp\left(\sum_{j,\ell} \tilde{w}_{j\ell} \log\left(p(y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\beta}_{k\ell}, \sigma_{k\ell}) \right) \right.$$

In the same manner, the posterior probability $\tilde{w}_{j\ell}$ is given by:

$$\tilde{w}_{j\ell} \propto \rho_\ell \exp\left(\sum_{i,k} \tilde{z}_{ik} \log\left(p(y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\beta}_{k\ell}, \sigma_{k\ell}) \right) \right).$$

M-step Given the previously computed posterior probabilities $\tilde{\mathbf{Z}}$ and $\tilde{\mathbf{W}}$, the M-step consists in updating , $\forall k, \ell$, the parameters of the model π_k, ρ_ℓ, and $\boldsymbol{\lambda}_{k\ell}$ maximizing $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \mathbf{\Omega})$. Using the computed quantities from step E, the maximization step (M-step) involves the following closed-form updates.

- Taking into account the constraints $\sum_k \pi_k = 1$ and $\sum_\ell \rho_\ell = 1$, it is easy to show that $\pi_k = \frac{\sum_i \tilde{z}_{ik}}{n} = \frac{\tilde{z}_{.k}}{n}$ and $\rho_\ell = \frac{\sum_j \tilde{w}_{j\ell}}{d} = \frac{\tilde{w}_{.\ell}}{d}$.
- The update of $\lambda_{k\ell}$ which is formed by $(\boldsymbol{\beta}_{k\ell}, \sigma_{k\ell})$, can be given by simple derivates of $F_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega})$ with respect to $\boldsymbol{\beta}_{k\ell}$ and $\sigma_{k\ell}$ respectively (See Appendix D). This leads to

$$\boldsymbol{\beta}_{k\ell} = \left(\sum_{i,j} \tilde{z}_{ik} \tilde{w}_{j\ell} y_{ij} \mathbf{x}_{ij} \right) \left(\sum_{i,j} \tilde{z}_{ik} \tilde{w}_{j\ell} \mathbf{x}_{ij} \mathbf{x}_{ij}^\top \right)^{-1},$$

and,

$$\sigma_{k\ell}^2 = \frac{\sum_{i,j} \tilde{z}_{ik} \tilde{w}_{j\ell} (y_{ij} - \boldsymbol{\beta}_{k\ell}^\top \mathbf{x}_{ij})^2}{\sum_{i,j} \tilde{z}_{ik} \tilde{w}_{j\ell}}.$$

The proposed algorithm for tensor data referred to as `VEM-LBRM` in Algorithm 8, alternates the two previously described steps Expectation-Maximization. At the convergence, a hard co-clustering is deduced from the posterior probabilities, and a regression model is deduced for ecah block $k\ell$.

Algorithm 8: `VEM-LBRM`

Input: $\mathcal{X}$, $\mathbf{Y}$, g, m.
Initialization $(\mathbf{z}, \mathbf{w})$ randomly, compute $\boldsymbol{\Omega}$
repeat
  E-Step
  - **Compute** $\tilde{z}_{ik}$ **using**
    $\tilde{z}_{ik} \propto \pi_k \exp\left(\sum_{j,\ell} \tilde{w}_{j\ell} \log\left(p(y_{i,j} | \mathbf{x}_{ij}, \boldsymbol{\beta}_{k\ell}, \sigma_{k\ell}) \right) \right)$
  - **Compute** $\tilde{w}_{j\ell}$ **using**
    $\tilde{w}_{j\ell} \propto \rho_\ell \exp\left(\sum_{i,k} \tilde{z}_{ik} \log\left(p(y_{i,j} | \mathbf{x}_{ij}, \boldsymbol{\beta}_{k\ell}, \sigma_{k\ell}) \right) \right)$

  M-Step
    Update $\boldsymbol{\Omega}$
until *convergence*;
return z, w, Ω

4.5 Experimental results

First, we evaluate the proposed `VEM-LBRM` on three synthetic datasets in terms of co-clustering and regression. We compare `VEM-LBRM` with some clustering and regression methods namely `Global model` which is a single multiple linear regression model performed on all observations and the following algorithms `K-means`, `Clusterwise`, `Co-clustering` and `SCOAL`. We retain two widely used measures to assess the quality of clustering, namely the Normalized Mutual Information (NMI) [Strehl and Ghosh, 2002] and the Adjusted Rand Index (ARI) [Liu et al., 2013b] (see section 1.1.5). On the other hand, we use RMSE (Root MSE) and MAE (Mean Absolute Error) metrics to evaluate the precision of prediction. While RMSE is a loss function which is suitable for Gaussian noises, MAE uses the absolute value

which is less sensitive to extreme values or outliers. The expression of RMSE and MAE can be written as follows:

$$RMSE = \sqrt{\frac{1}{|\mathcal{R}|} \sum_{(i,j)\in(I,J)} (r_{ij} - \hat{r}_{ij})^2},$$

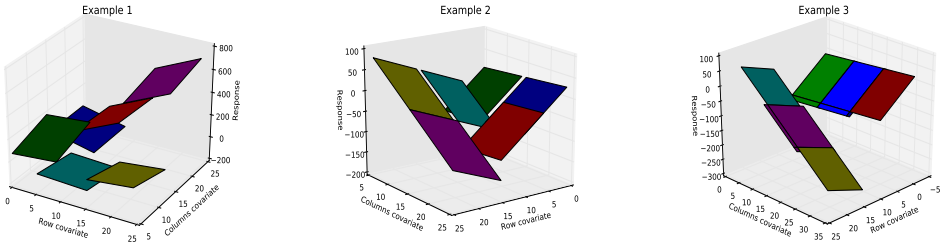
and,

$$MAE = \frac{1}{|\mathcal{R}|} \sum_{(i,j)\in(I,J)} \left| r_{ij} - \hat{r}_{ij} \right|.$$

Secondly, we present the results of `VEM-LBRM` on a small dataset, as an illustrative example. Finally, we propose to apply `VEM-LBRM` for recommender systems application, using five real-word datasets. Through this evaluation, we aim to demonstrate the impact of covariates information on interpretation and improvement of clustering and regression results, and the benefit of the joint co-clustering and regression learning .

4.5.1 Simulation study

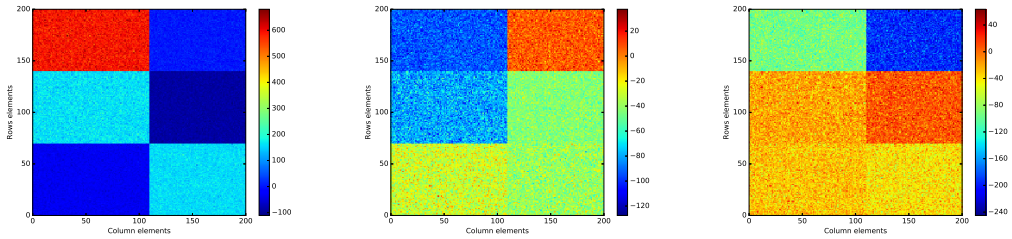
We generated tensor data $\mathcal{X}$ with size $200 \times 200 \times 2$ according to Gaussian model per block. In the simulation study, we considered three scenarios by varying the regression parameters — the examples are generated with different regression collinearity and different co-clusters structure complexity. The parameters for each example are reported in Tables 4.1. In Figures 4.5 and 4.6 are depicted the true regression plans and the true simulated response matrix $\mathbf{Y}$.



(A) (B) (C)

FIGURE 4.5: Synthetic data: True regression plans according to the chosen parameters.

In our illustrations, we consider co-clustering and regression challenges. All metrics concerning rows and columns are computed by averaging on ten random training, and testing data split using an 80% vs. 20% of training and validation data. Thereby, we compare `VEM-LBRM` with Global model (which is a multiple linear regression), `K-means`, `Clusterwise` by reshaping the tensor to matrix with size $N \times v$ where $N = n \times d$. On the other hand, the `VEM` algorithm for co-clustering is applied on response matrix $\mathbf{Y}$. Furthermore, for clustering algorithms, the RMSE, MAE, and R-squared (R^2 Avg.) are computed by applying linear regression on each obtained co-cluster. In Table, 4.2 are reported the performances for all algorithms. The missing values '-' represent measures that cannot be computed by the corresponding algorithms. From these comparisons, we observe that whether



(A) (B) (C)

FIGURE 4.6: Synthetic data: True co-clustering according to the chosen parameters.

TABLE 4.1: Parameters generation for synthetic data.

Dataset	Example 1		Example 2		Example 3	
	$\boldsymbol{\pi} = [0.35, 0.35, 0.3]$, $\boldsymbol{\rho} = [0.55, 0.45]$					
σ	$\sigma = 5$		$\sigma = 7$		$\sigma = 7$	
$\boldsymbol{\Sigma}$	$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$		$\boldsymbol{\Sigma} = \begin{bmatrix} 2 & 0.3 \\ 0.3 & 2 \end{bmatrix}$		$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$	
Co-clusters	$\boldsymbol{\beta}_{k\ell}$	$\boldsymbol{\mu}_{k\ell}$	$\boldsymbol{\beta}_{k\ell}$	$\boldsymbol{\mu}_{k\ell}$	$\boldsymbol{\beta}_{k\ell}$	$\boldsymbol{\mu}_{k\ell}$
Cluster (1,1)	[1, -10, 1]	[5,20]	[1, -10, 1]	[5,20]	[1, -10, 1]	[5,20]
Cluster (1,2)	[10, 4, 13]	[5,10]	[1, -10, 1]	[5,10]	[1, -10, 1]	[5,10]
Cluster (2,1)	[3, 20, -2]	[10,20]	[1, -10, 1]	[10,20]	[1, -10, 1]	[5,30]
Cluster (2,2)	[-5, -2, -6]	[10,10]	[7, 5, -10]	[10,10]	[7, 5, -10]	[20,10]
Cluster (3,1)	[-10, 20, 10]	[20,20]	[7, 5, -10]	[20,20]	[7, 5, -10]	[20,20]
Cluster (3,2)	[7, 5, -10]	[20,10]	[7, 5, -10]	[20,10]	[7, 5, -10]	[20,30]

the block structure is easy to identify or not, the ability of `VEM-LBRM` to outperform other algorithms. More precisely, `VEM-LBRM` does a better job than `SCOAL` and `Co-clustering` algorithms, especially in the third example, which presents high collinearity between covariables.

4.5.2 Illustrative example

To illustrate the interest of the proposed model, we present a simple and illustrative real-world dataset. The study concerns six types of strawberry farmers in Uruguay, namely Festival, Yvahé, Yurí, Guenoa, L20.1, and K31.5. The dataset consists of 116 consumers. They evaluate strawberries based on 16 attributes (size, sweet, odour, shape, etc.). The evaluation is based on a 9-point rating scale. The used data table can be download from the following web site[1].

We realize 30 runs with 80-20% of train and test sample. We select the ten better runs based on the log-likelihood, and a consensus between obtained clusters is achieved using a `Cluster_Ensembles` package in Python. In fact, we applied a consensus on row partitions using the ten best runs in terms of log-likelihood, and same process is a applied for

[1] http://crcpress.com/product/isbn/9781466566293

TABLE 4.2: (co)-clustering and prediction: "mean" and "standard deviation" in parentheses.

Examples	Algorithms	Regression					Clustering			
		RMSE		MAE		R^2	ARI		NMI	
		Training	Test	Training	Test	Avg.	Row	Col	Row	Col
Example1	Global Model	164.38	164.05	145.29	145.05	0.46	-	-	-	-
		(0.03)	(0.49)	(0.08)	(0.71)	(0.0)	-	-	-	-
	K-means	49.62	49.51	34.86	34.91	0.8	0.61	-	0.49	-
		(60.2)	(67.48)	(33.56)	(35.79)	(0.02)	0.02	-	0.03	-
	Clusterwise	154.57	154.47	127.77	127.93	0.52	0.07	-	0.01	-
	($g = 3$)	(0.01)	(0.36)	(0.03)	(0.45)	(0.0)	0.0	-	0.0	-
	Co-clustering	10.86	10.83	7.29	7.29	0.88	0.84	1.0	0.71	1.0
	($g = 3$)	(14.76)	(14.36)	(4.67)	(4.59)	(0.0)	0.01	0.0	0.04	0.0
	SCOAL	14.99	14.92	10.45	10.41	0.99	0.91	1.0	0.84	1.0
	($g = 3, m = 2$)	(207.56)	(208.91)	(89.48)	(90.55)	(0.0)	0.01	0.0	0.04	0.0
	VEM-LBRM	**7.1**	**7.06**	**5.29**	**5.26**	**0.99**	**0.95**	**1.0**	**0.92**	**1.0**
	($g = 3, m = 2$)	**(17.71)**	**(16.86)**	**(6.8)**	**(6.32)**	**(0.0)**	**(0.01)**	**(0.0)**	**(0.03)**	**(0.0)**
Example2	Global Model	29.15	29.21	24.64	24.68	0.34	-	-	-	-
		(0.04)	(0.15)	(0.04)	(0.12)	(0.0)	-	-	-	-
	K-means	10.43	10.49	7.73	7.77	0.71	0.56	-	0.45	-
		(0.25)	(0.24)	(0.17)	(0.16)	(0.01)	0.0	-	0.0	-
	Clusterwise	18.54	18.62	11.33	11.38	0.73	0.15	-	0.16	-
	($g = 3$)	(0.09)	(0.27)	(0.06)	(0.14)	(0.0)	0.0	-	0.0	-
	Co-clustering	7.5	7.49	5.89	5.9	0.8	0.95	1.0	0.94	1.0
	($g = 3$)	(1.35)	(1.38)	(0.82)	(0.86)	(0.07)	0.14	0.0	0.17	0.0
	SCOAL	12.63	12.69	8.75	8.81	0.81	0.97	1.0	0.94	1.0
	($g = 3, m = 2$)	(12.57)	(12.81)	(7.38)	(7.58)	(0.35)	0.1	0.0	0.17	0.0
	VEM-LBRM	**6.99**	**6.99**	**5.57**	**5.57**	**0.96**	**1.0**	**1.0**	**1.0**	**1.0**
	($g = 3, m = 2$)	**(0.01)**	**(0.04)**	**(0.01)**	**(0.02)**	**(0.0)**	**(0.0)**	**(0.0)**	**(0.0)**	**(0.0)**
Example3	Global Model	45.38	45.24	38.33	38.21	0.49	-	-	-	-
		(0.06)	(0.24)	(0.07)	(0.26)	(0.0)	-	-	-	-
	K-means	10.47	10.41	7.44	7.42	0.83	0.54	-	0.45	-
		(1.73)	(1.74)	(1.08)	(1.08)	(0.08)	0.01	-	0.01	-
	Clusterwise	23.09	23.18	12.09	12.15	0.87	0.09	-	0.09	-
	($g = 3$)	(1.84)	(2.02)	(1.23)	(1.29)	(0.02)	0.0	-	0.0	-
	Co-clustering	9.48	9.39	6.98	6.93	0.73	0.74	1.0	0.7	1.0
	($g = 3$)	(0.16)	(0.22)	(0.01)	(0.02)	(0.02)	0.04	0.0	0.08	0.0
	SCOAL	27.32	27.14	16.82	16.73	0.57	0.98	1.0	0.96	1.0
	($g = 3, m = 2$)	(41.97)	(41.83)	(24.13)	(24.16)	(0.93)	0.07	0.0	0.12	0.0
	VEM-LBRM	**7.21**	**7.21**	**5.71**	**5.71**	**0.99**	**0.98**	**1.0**	**0.96**	**1.0**
	($g = 3, m = 2$)	**(0.68)**	**(0.7)**	**(0.42)**	**(0.42)**	**(0.0)**	**(0.07)**	**(0.0)**	**(0.12)**	**(0.0)**

column partitions. We update the parameters of models based on the consensus clustering results. Figure 4.7 represents the true and predicted rating matrix for the test set using the described process. We can see that the predicted values are very close to the true ones. The

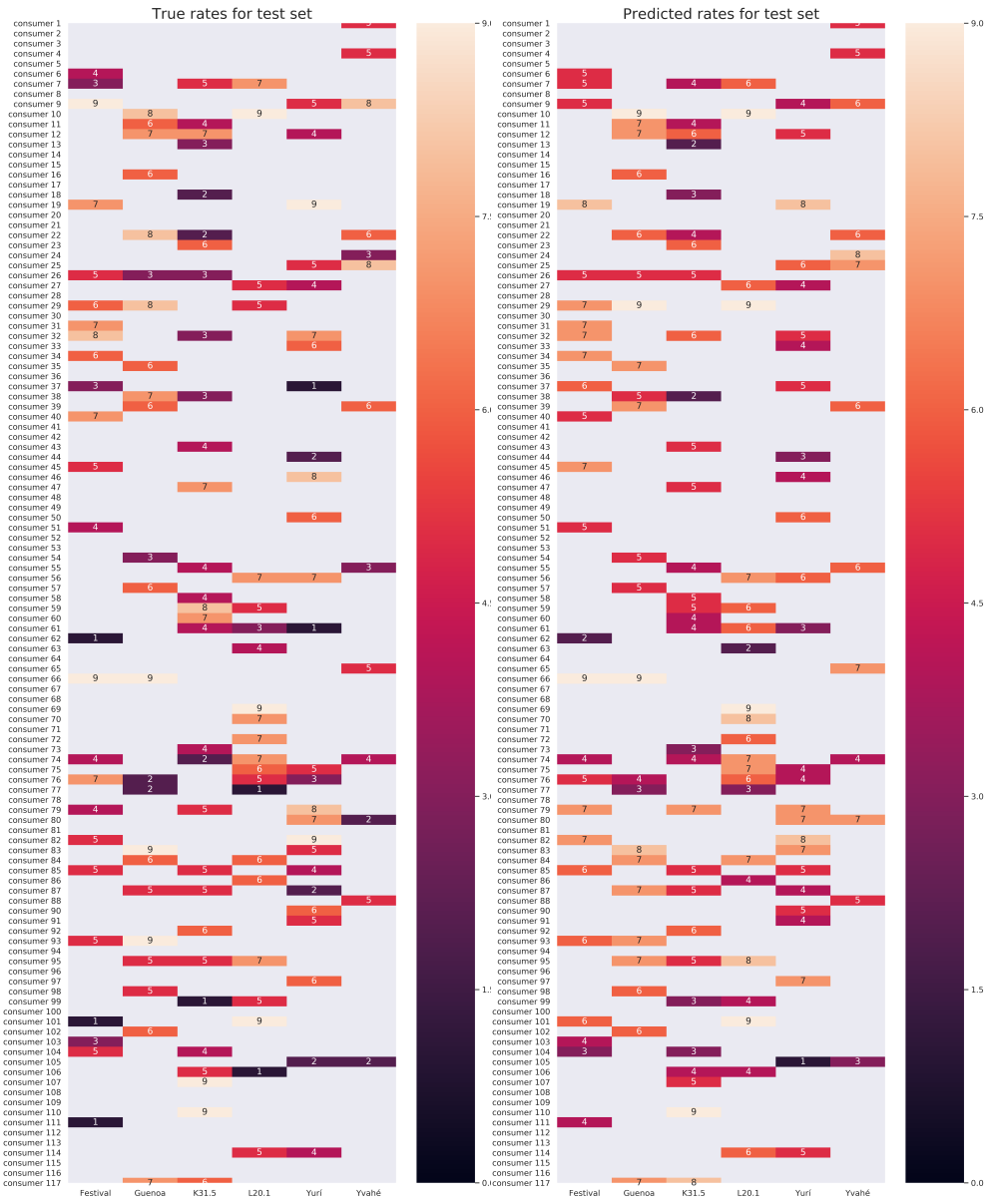


FIGURE 4.7: Obtained results on Strawberry dataset.

Mean Absolute Error (MAE) for the test set is 1.31. Noting that the rating scale is from 1 to 9, the MAE value informs us about the generalization capacity of our model. In fact, the obtained error equal to 1.3 in the test phase seems promising and allows us to conclude that our model fails in predicting rates with error, on avarage, of one point.

On the other hand, we plot in figure 4.8 the mean of attributes vector for each co-clusters (consumer/strawberry type). We find 3 clusters of strawberry cultivars and 3 clusters of consumers. The strawberry clusters highlight that Yvahé, Yurí has a similar characteristic with irregular shape and sour strawberries. Festival, L20.1, and K31.5 belong to the same cluster and characterized by firm texture and tasteless. Finally, Guenoa is a particular strawberry

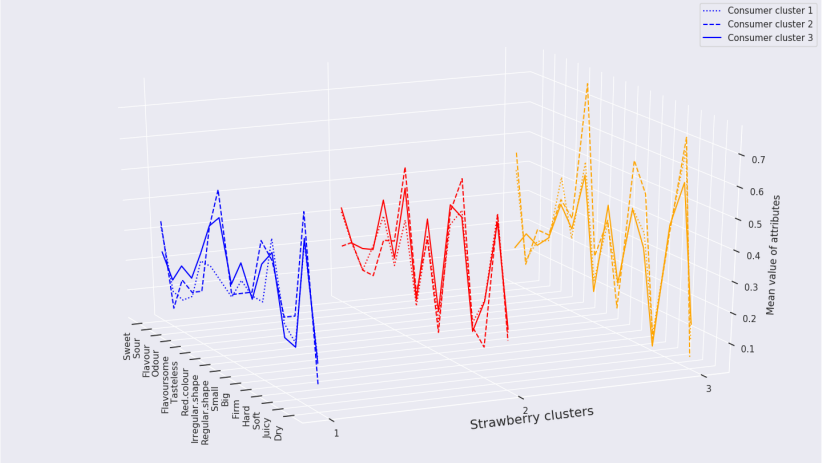


FIGURE 4.8: Representation of mean of co-clusters.

type with some specific characteristics such as a high value of Jucy and red color attributes (with more than 70%), and high sweet (more than 45% of a consumer).

4.5.3 Recommender system application

To show the benefits of our approach, we select five popular real-world datasets for recommender systems, namely Movielens100K, FilmTrust, Yahoo! Movies, Yahoo! Music and Jester.

- **Movielens100K**[2]**.** The Movielens100k database consists of 100,000 ratings of 943 users and 1682 movies (from 1 to 5). Each user has rated at least 20 movies, then we construct users-movies rating matrix (943×1682) and assign 0 to movie without rating. Furthermore Movielens100K dataset includes 23 user covariates including age, gender and 21 employment status. Furthermore, we have in our disposal 19 covariates related to movie genres, considering that movie may belong to one or more genres.

- **FilmTrust**[3]**.** This rating dataset is obtained from the FilmTrust website. Unlike the Movielens100K datasets, the covariates about users and items are not available.

- **Yahoo! Movies**[4]**.** This dataset is developed by Yahoo! Research for research on classification and recommender systems. It contains 7,642 users and 11,915 movies. Only the users and movies with more than 20 interactions are selected. Similar to FilmTrust the covariates about users and items are not available.

- **Yahoo! Music**[4]**.** Yahoo! Music dataset contains rating data of 15,400 users and 1,000 songs. Likewise Yahoo! Movies dataset, we include only users and movies with more than 20 interactions.

[2]http://grouplens.org/datasets/movielens/
[3]http://www.librec.net/datasets.html
[4]https://webscope.sandbox.yahoo.com/catalog.php?datatype=r

- **Jester**[5]**.** The Jester dataset was built from online jokes recommender system. The ratings data contains 24,983 users and 100 jokes. Each user rated no less than 36 jokes.

Movielens100K is the only dataset with available user and item covariates. For other datasets, six features derived from the data matrix $\mathbf{Y}$ were used. The features represented user covariates are:

- Number of items rated by each user.
- Average of the ratings given by each user.
- Variance of the ratings given by each user.

On the other hand, the features represented item covariates are:

- Number of users that rated each item.
- Average of the ratings obtained from each item.
- Variance of the ratings obtained from each item.

Table 4.3 provides some information about the five datasets, namely the number of users and items, the number and scale of ratings, rating matrix density, and the number of covariates.

TABLE 4.3: Description of Datasets.

Characteristic	Datasets				
	Movielens 100K	**FilmTrust**	**Yahoo! Movies**	**Yahoo! Music**	**Jester**
Users	943	1,508	4,385	4,748	24,983
Items	1,682	2,071	4,339	1,000	100
Ratings	100,000	35,497	169,767	196,150	705,378
Ratings-scale	[1,5]	[0.5,4]	[1,5]	[1,5]	[-10,10]
Density	6.3%	1.14%	0.89%	4.13%	28.23%
Users covariate	Yes	No	No	No	No
Items covariates	Yes	No	No	No	No

From these datasets, we aim to measure the impact of covariates on improving prediction results. The average of RMSE, MAE and R^2 for various algorithms are computed using a 5-fold cross-validation method. We use the same number of row and column clusters for the all datasets; $g = m = 4$. Further, we use the Recall@k, Precision@k, and F-measure@k measures to evaluate the proposed algorithm in terms of recommendation; k is the number of top items in the recommendation list.

- **Precision@k**:For each user the Precision@k denotes the proportion of good items in his/her top-k recommendation list. To evaluate an entire CF system we compute the average Precision@k over all users.

[5]http://www.ieor.berkeley.edu/ goldberg/jester-data/

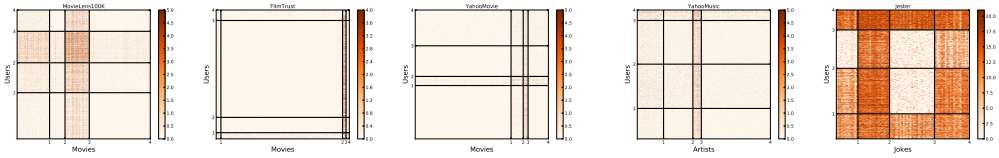
- **Recall@k**:The Recall@k for a user is the proportion of good items, in the user's top-k recommendation list, from the number of relevant held-out items for that user. As for the above measures, we can compute the average Recall@k over all users to evaluate an entire model.

- **F-measure@k**:The F-measure@k combines both Precision@k and Recall@k into a single measure to find a trade off. The F-measure@k can be computed as a harmonic mean of recall and precision measures:

$$F - measure@k = \frac{Recall@k \times Precision@k}{Recall@k + Precision@k}$$

To evaluate recommendation results, We use the most popular k values to compute Precision@k, Recall@k, namely k equals to 3, 5, and 10.

In figure 4.9, we show the reorganization of rating matrix for all datasets using the clustering results obtained by VEM−LBRM. We can see that the proposed algorithm tend to find homogeneous block.



(A) Movilens (B) FilmTrust (C) Yahoo! Movies (D) Yahoo! Music (E) Jester

FIGURE 4.9: Co-clustering results on rating matrix obtained by VEM−LBRM on all datasets.

What is the impact of covariates on regression results? The objective of this part, is to show the impact of covariates on improving the prediction results. In fact, we use the five datasets, an we apply the proposed algorithm VEM−LBRM, with user and items covariates separately and also considering both covariates. In table 4.4, we report the obtained results in terms of RMSE and MAE of training and test sets. We also compute the R^2 Avg. for regression models.

In most cases, VEM−LBRM using only user's covariates, obtained better results than using item's covariates. This allows us to support the assumption[6] presented in [Ricci et al., 2011], where the authors explain that user-based model, are more effective than item-based model for datasets with a small number of users (almost equal to number of items).

Also, we can see that using both covariates of users and items simultaneously, allow us to achieve the lower RMSE and MAE errors and higher R^2 Avg. This support our assumption, in fact, co-cluster of users having same profiles and interested by same item's types have the same rating behaviors.

[6]In cases where the number of users is much greater than the number of items, such as large commercial systems like Amazon.com, item-based methods can therefore produce more accurate recommendations [Fouss et al., 2007, Last.fm, 2009]. Likewise, systems that have fewer users than items, e.g., a research paper recommender with thousands of users but hundreds of thousands of articles to recommend, may benefit more from user-based neighborhood methods [Good et al., 1999].

TABLE 4.4: Covariate impact on Datesets using the proposed VEM−LBRM algorithm.

Datasets	Covariates	RMSE Training	RMSE Test	MAE Training	MAE Test	R^2 Avg.
MovieLens100K	Users covariate	1.059	1.087	0.856	0.871	0.115
		(6e-06)	(8.9e-05)	(1.1e-05)	(5.1e-05)	(1.6e-05)
	Items covariate	1.061	1.088	0.856	0.871	0.111
		(3e-06)	(2.6e-05)	(5e-06)	(1.6e-05)	(7e-06)
	Users and Items	**1.041**	**1.072**	**0.838**	**0.856**	**0.145**
	covariate	**(2e-06)**	**(1.7e-05)**	**(3e-06)**	**(1.6e-05)**	**(4e-06)**
FilmTrust	Users covariate	0.782	0.856	0.602	0.631	0.277
		(7e-06)	(0.0006)	(5e-06)	(0.0001)	(1.5e-05)
	Items covariate	0.86	0.921	0.674	0.7	0.126
		(5e-06)	(0.0007)	(5e-06)	(0.0002)	(8e-06)
	Users and Items	**0.731**	**0.807**	**0.56**	**0.588**	**0.367**
	covariate	**(4e-06)**	**(0.0010)**	**(3e-06)**	**(0.0002)**	**(3e-06)**
Yahoo! Movies	Users covariate	1.04	1.111	0.775	0.806	0.216
		(1e-06)	(0.0003)	(1e-06)	(6e-05)	(3e-06)
	Items covariate	1.032	1.098	0.764	0.793	0.228
		(2e-06)	(6.7e-05)	(1e-06)	(1.9e-05)	(3e-06)
	Users and Items	**0.936**	**1.013**	**0.68**	**0.71**	**0.365**
	covariate	**(1e-06)**	**(0.0005)**	**(1e-06)**	**(8.7e-05)**	**(2e-06)**
Yahoo! Music	Users covariate	1.172	1.185	0.907	0.916	0.429
		(1e-06)	(1.4e-05)	(1e-06)	(1.4e-05)	(1e-06)
	Items covariate	1.347	1.368	1.123	1.136	0.246
		(1e-06)	(1e-05)	(2e-06)	(8e-06)	(2e-06)
	Users and Items	**1.145**	**1.159**	**0.884**	**0.892**	**0.455**
	covariate	**(1e-06)**	**(8e-06)**	**(1e-06)**	**(6e-06)**	**(1e-06)**
Jester	Users covariate	4.364	4.366	3.461	3.463	0.305
		(1e-06)	(1.4e-05)	(2e-06)	(7e-06)	(0.0)
	Items covariate	4.826	4.828	3.966	3.969	0.15
		(0.002)	(0.0019)	(0.002)	(0.0018)	(0.0002)
	Users and Items	**4.247**	**4.249**	**3.352**	**3.354**	**0.342**
	covariate	**(3e-06)**	**(4e-06)**	**(3e-06)**	**(1e-06)**	**(0.0)**

How does VEM−LBRM improve the precision of the recommendation? In this second part of experimentation, we evaluate VEM−LBRM in terms of recommendation performances. We compare VEM−LBRM with co−clustering and NMF algorithms.

In figures 4.10 and 4.11, we report recommendation results through Recall@k, Precision@k, and F-measure@k for NMF, co−clustering, and VEM−LBRM. We use the implementation of NMF and co−clustering available on *Surprise*[7] package. In terms of precision and recall measures, VEM−LBRM does, in almost all cases, a better job than NMF and co−clustering for all datasets.

[7] http://surpriselib.com/

However, in the light of Recall@5 and Recall@10 for Jester and YahooMusic, `Co-clustering` is more effective; this can be explained by the cold-start problem, which occurs in sparse datasets.

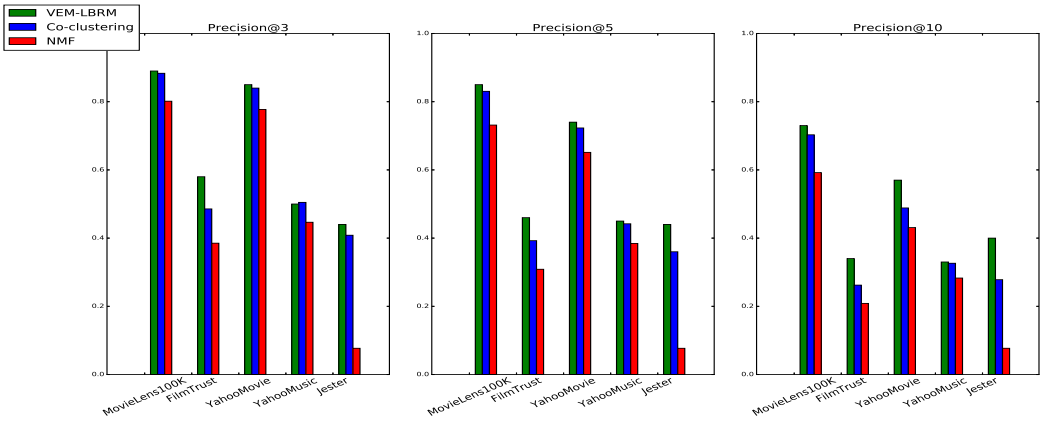


FIGURE 4.10: Precision of k-top recommendations using `NMF`, `co-clustering` and `VEM-LBRM` for all datasets.

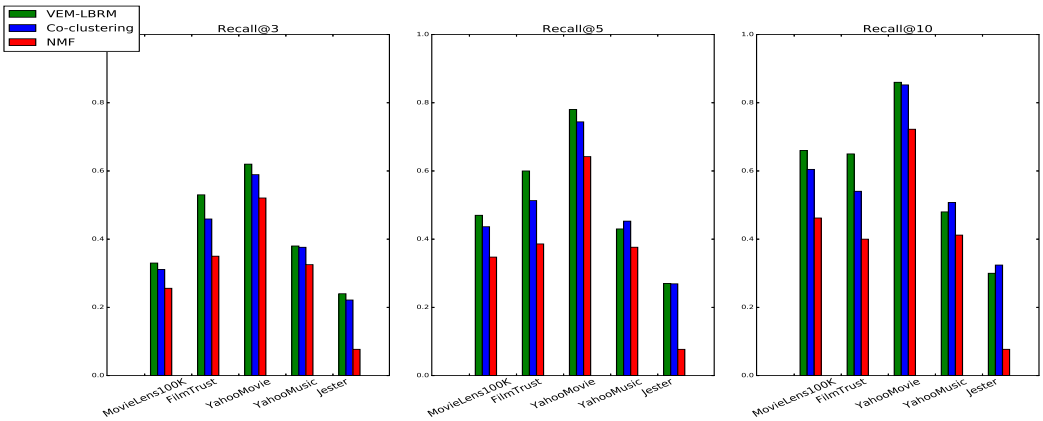


FIGURE 4.11: Recall of k-top recommendations using `NMF`, `co-clustering` and `VEM-LBRM` for all datasets.

In Table 4.5 are reported the performances of `VEM-LBRM`, `NMF`, and `Co-clustering` proposed by Thomas and Merugu [2005] for all datasets, in terms of F-measure@3, F-measure@5, and F-measure@10. F-measure represents a trade-off between precision and recall measures. `VEM-LBRM` achieves better results with higher values of F-measure@k for the five datasets. Furthermore, we notice that the percentage of improvement of `VEM-LBRM` comparing to the best results among `NMF` and `co-clustering`, can reach 18%. For FilmTrust, which is one of the most sparse datasets, we reach a greater improvement. On the other hand, for YahooMusic, there is any improvement using `VEM-LBRM` comparing to `co-clustering`.

TABLE 4.5: F-measure of k-top recommendations using NMF, co-clustering and VEM-LBRM for all datasets.

Datasets	Measures	NMF	co-clustering	VEM-LBRM	Improve(%)
MovieLens100K	F-measure@3	0.33	0.40	**0.41**	**2.4%**
	F-measure@5	0.40	0.49	**0.51**	**3.9%**
	F-measure@10	0.44	0.55	**0.58**	**5.2%**
FilmTrust	F-measure@3	0.32	0.41	**0.48**	**14.6%**
	F-measure@5	0.30	0.39	**0.46**	**15.2%**
	F-measure@10	0.24	0.31	**0.38**	**18.4%**
YahooMovie	F-measure@3	0.56	0.63	**0.64**	**1.6%**
	F-measure@5	0.58	0.66	**0.68**	**2.9%**
	F-measure@10	0.47	0.55	**0.57**	**3.5%**
Jester	F-measure@3	0.07	0.27	**0.28**	**3.6%**
	F-measure@5	0.07	0.29	**0.30**	**0.7%**
	F-measure@10	0.07	0.29	**0.31**	**6.5%**
YahooMusic	F-measure@3	0.35	**0.40**	**0.40**	**0.0%**
	F-measure@5	0.36	**0.42**	**0.42**	**0.0%**
	F-measure@10	0.31	**0.37**	**0.37**	**0.0%**

4.6 Conclusion

In this chapter, we proposed an extension of LBM to tensor data, aiming both tasks: co-clustering and prediction. The proposed model refereed to as LBRM gives rise to a variational EM algorithm for co-clustering and prediction referred to as VEM-LBRM. This algorithm, which can be viewed as a *co-clusterwise* algorithm, can easily deal with sparse data. Empirical results on synthetic and real-world datasets show that VEM-LBRM gives encouraging results than some algorithms devoted to one or both tasks simultaneously. Furthermore, we evaluated VEM-LBRM in terms of recommendation performances using various measures such as Recall@k, Precision@k, and F-measure@k. We notice that VEM-LBRM improves the results of recommendation in most cases.

It is a known fact that multiple linear regression suffers from the over-fitting and multi-collinearity. In the literature, several variants of regression were proposed to overcome these drawbacks. Ridge and Lasso's regressions, for example, are simple techniques to handle with collinearity, prevent over-fitting, and deal with outliers. Their integration in LBRM can be a good way to deal with very sparse datasets containing collinear covariates.

Chapter 5

Using Tensor Analysis for Original Applications

This chapter is dedicated to evaluate our algorithms and demonstrate the strong points of proposed approaches on real-world applications. In section 5.1, as part of *CIFRE* thesis, we focus on waste management applications; the aim is to show the advantages of the proposed algorithms and their capacity in improving recommendations and optimizations of waste management. Furthermore, in section 5.2, we are going to apply some of these algorithms for the EGC challenge to analyze the evolution of the EGC conference. The obtained results for all applications will be presented and interpreted.

5.1 Waste management applications

In the past two decades, France has been engaged in the challenge of transition to a circular economy model, a necessary action for ecological development. Given the limited resources of our planet's ecosystem, it is essential to quit the linear model of "take-make-consume-throw" and progress towards a circular economy. This implies curbing land-filling and promoting recycling, reuse, and re-manufacturing [Bourguignon, 2014].

Five tonnes of waste per capita are generated every year in the European Union (EU), mostly from the construction and mining sectors. Thus, waste management can have adverse effects on the environment, climate, and human health. In 2016, total waste production in France amounted to 323.4 million tonnes. Few data are available on the cost of waste management. Data from the French Ministry for Ecology estimate that, in 2010, the total cost of waste management in France was €377/tonne. The French agency for environment and energy (ADEME) shows varying average net costs of treatment depending on treatment method: €180/tonne to landfill residual municipal waste, €203/tonne to incinerate residual municipal waste and €343/tonne to treat recyclable waste [Bourguignon, 2015].

In France, the transition project was implemented by the Energy Transition Law (*loi de transition énergétique pour la croissance verte*) and was reaffirmed by the plan of reducing and enhancing waste energy costs by 2025, published in December 2016. Therefore, the recent roadmap for the circular economy (FREC[1]) announced the modernization of the legislation providing an adaptation to the challenge of circular economy transition. In fact, in 2016, the French government passed a law known as *5 flux* obliging companies and businesses to sort their waste into at least five different waste types (paper/cardboard, metal,

[1] https://www.ademe.fr/feuille-route-collecte-tri-recyclage-valorisation-dechets

plastic, glass, and wood). Also, the TGAP (Taxe Générale sur les Activités Polluantes) tax is paid by companies and industries producing large quantities of waste (construction, retail, etc.). The law provides that the TGAP would increase to €54/tonne for landfilling waste until reaching a cost of €65/tonne in 2025. The incineration TGAP would increase to €20/tonne in 2021, and up to €25 in 2025 [Turchet, 2018].

Artificial intelligence (AI) and digital innovation are paving the way for a new generation of sorting centers. Currently, in France, robots allow more than 60% of waste sorting per hour comparing to a human being. The implementation of such type of intelligent sorting robot, equipped with learning mechanisms for recognizing the different waste types, represents a significant challenge on which recycling centers rely–noting that digital and technological innovations are profitable for other domains of energy and environment sector.

Data science and data analysis are becoming essential tools for optimizing waste management. The deployment of sensors is gradually generalizing the concept of "connected bin". It allows us to collect a significant amount of data or "big data" in real-time, such as the rate of filling of bins, composition of waste, and waste quantity, which can help to optimize the management of the waste collection and improve sorting performance.

Trinov is a young innovative company that fits in this context and combines two areas of expertise: the first on waste management, and the second in information technology, data mining, and machine learning algorithms. As both a consultant and a technology provider, *Trinov* has the ambition to become a key player in waste management by creating solutions, tools, and intelligent algorithms for optimizing waste management. Aiming to develop high-performance decision support tools for waste management, *Trinov* developed a set of tools for collecting and analyzing data. These tools provide quantitative and qualitative data (volume of waste production, the type of waste containers, geolocation data, etc.) using connected objects and data provided by the waste operators. The objective of this thesis is to propose models of optimization and recommendation adapted to customers, and that ensures the improvement of waste management, including different tasks such as the optimization of waste collection, the recommendation of waste containers' type, the estimation of the number of waste collections, etc.

To this end, three projects related to waste management were developed. The first one is about the optimization of waste collection number and the recommendation of containers' type. This issue was developed for retails but also for other sectors such as hospitals, public transportation companies, etc. The aim is to create a high-performance recommendation system that allows us to predict the number of collections per month and provide recommendations about the container's type. The purpose of these recommendations is to minimize waste management costs. The second project concerns waste collection. We propose efficient waste collection algorithms combining `TSPLBM` and genetic algorithm to optimize waste collection and significantly reduce costs. The last work is about the markdown analysis. Actually, a markdown in retails is a group of products that are broken, outdated, stolen, not suitable for consumption, etc. The objective of this part is to analyze the markdown behaviors considering stores, product categories and causes.

5.2 Analysis of EGC conference evolution

The EGC conference is one of the most popular French conferences attracting a large number of researchers each year. For the 20th edition, the conference proposed a challenge[4] for analyzing and predicting the evolution of the conference since 2001.

In the sequel, we propose a multi-dimensional analysis from different data sources (see figure 5.10) to extract relevant information. To do this, we first performed a data preprocessing and constructed three-way tensors, allowing us to combine different information. The three main contributions of this work are (i) the extraction of the topics from the papers published in the EGC conference and the analysis of the temporal aspect of topic evolution (ii), the analysis of authors' communities, (iii) the recommendation of reviewers for the *lecture committee*.

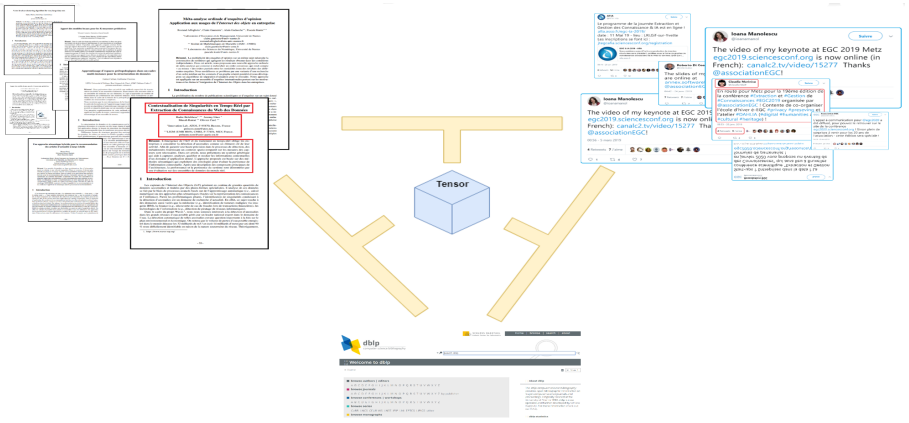


FIGURE 5.10: Different data sources for the EGC challenge.

5.2.1 Data preprocessing and description

To analyze the history of the EGC conference, we have at our disposal the list of all papers, its contents, and the PDF version of all papers. We performed a data preprocessing and selected the most relevant variables for our study. An exogenous data was also introduced to help us with the interpretation of results.

Papers and their content

After the preprocessing phase, we extracted a set of relevant information:

- Titles of papers.
- Abstract of papers.
- Authors list for each paper.
- The affiliation of authors.
- The list of references cited by each paper, extracted from the PDF version of papers.

[4] https://www.egc.asso.fr/manifestations/defi-egc/defi-egc-2020-20-ans-dhistoire-pour-quel-avenir.html

In this study, we focus only on the papers without missing data (ie, 1096 papers). Several data matrices have been constructed:

- Let $\mathcal{T}$ be a documents-terms matrix based on the titles of papers (documents); each cell $\mathcal{T}[i,j]$ represents the occurrence of the word j in the paper i.
- Let $\mathcal{R}$ be a documents-terms matrix based on the abstracts of papers, it is constructed in the same way as the matrix $\mathcal{T}$.
- Let $\mathcal{A}$ be a documents-authors matrix; each cell $\mathcal{A}[i,j]$ is equal to 1 if j is author of paper i, and 0 otherwise.
- Let $\mathcal{F}$ be a documents-references matrix; each cell $\mathcal{F}[i,j]$ is equal to 1 if the reference j has been quoted in the paper i, and 0 otherwise.
- Let $\mathcal{H}$ be a authors-affiliations matrix; each cell $\mathcal{H}[i,j]$ is equal to 1 if the author i belongs to the institution j, and 0 otherwise.
- Let $\mathcal{B} = \mathcal{A}^T\mathcal{T}$ be a authors-terms matrix constructed from the binarized matrix $\mathcal{T}$ and $\mathcal{A}$; each cell $\mathcal{B}[i,j]$ represents the number of times that the term j was used by the author i.

Exogenous data extracted from the Web

We extracted information using the DBLP API. We consider all the information regarding previous publications for all authors, including titles of their publications. To enrich our analysis of authors' communities, the *sex* variable of authors was scraped from the web.

5.2.2 Topic modeling of papers

In order to analyze the topics, we have built four graphs representing different relationships between documents:

1. The co-terms title matrix is constructed from the binarized documents-terms matrix of titles and is computed by $\mathcal{T}\mathcal{T}^T$.
2. The co-terms abstract matrix is constructed in the same way as the matrix of the co-terms title but from the matrix documents-terms of the abstracts such as $\mathcal{R}\mathcal{R}^T$.
3. The co-authors matrix is constructed from the documents-authors matrix representing the number of similar authors who contributed to the paper by computing $\mathcal{A}\mathcal{A}^T$.
4. The co-references matrix is constructed from documents-references matrix $\mathcal{F}$, where each cell represents the number of references in common between two papers. This matrix is obtained by $\mathcal{F}\mathcal{F}^T$.

Using these four relationships between papers, we construct the $Papers \times Papers \times Relationships$ tensor, with size $1096 \times 1096 \times 4$. Then, we applied the `TSPLBM` algorithm on this tenor using a number of clusters equals to 8 based on the modularity measure. Figure 5.11 represents the reorganization of nodes of the four graphs using the partitioning obtained by `TSPLBM`.

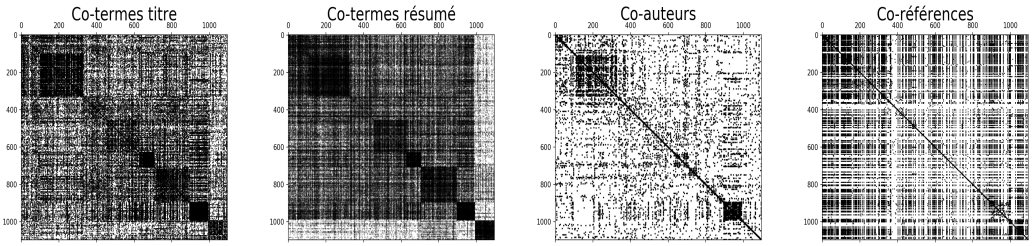


FIGURE 5.11: Reorganization of adjacency matrices of papers.

What are the characteristics describing the discovered topics? From the obtained document clusters (topics), and using the $\mathcal{T}$ documents-terms matrices and the $\mathcal{A}$ documents-authors matrix, we constructed the $\mathcal{D}$ topics-terms and $\mathcal{G}$ topics-authors matrices respectively. The topics-terms matrix represents the number of times that the term has been mentioned in the topic. The topics-authors matrix reports the number of papers that an author has published in this topic.

We applied a correspondence analysis (CA) [Benzecri, 1973] on the topic-term and topics-authors matrices. The choice of visualization by CA is justified by the Poissonian model of latent blocks; for more details, see [Govaert and Nadif, 2018]. The obtained results are illustrated in the figure 5.12 (respectively 5.13). The strong point of `TSPLBM` is its ability to represent objects (papers) using multiple views (terms, authors, references). Figure 5.14 presents the frequencies of terms for each cluster of documents (topic). We can describe the eight topics as follows:

- Topic 1: Unsupervised learning approaches.
- Topic 2: Supervised/Unsupervised learning methods and all issues related to machine learning.
- Topic 3: Knowledge extraction.
- Topic 4: Graph knowledge bases and ontologies.
- Topic 5: Association rules.
- Topic 6: Semantic Web.
- Topic 7: Patterns extraction.
- Topic 8: Data mining, with a majority of English papers. Noting that this topic is most distinguished from others in figure 5.12.

In figure 5.13, we can observe that each topic is characterized by group of researchers who contribute significantly on this topic in EGC conference. For topic 2 dealing with machine learning, we observe authors like *Vincent Lemaire*, *Marc Boulle*. For topic 7 dealing with patterns detection, we observe authors such as *Marc Plantvit* and *Céline Robardet*. Finally, topic 4 dealing with association rules is illustrated by authors such as *Florence Sedges*, *Marc Le Goc*, and *Philippe Bouché*.

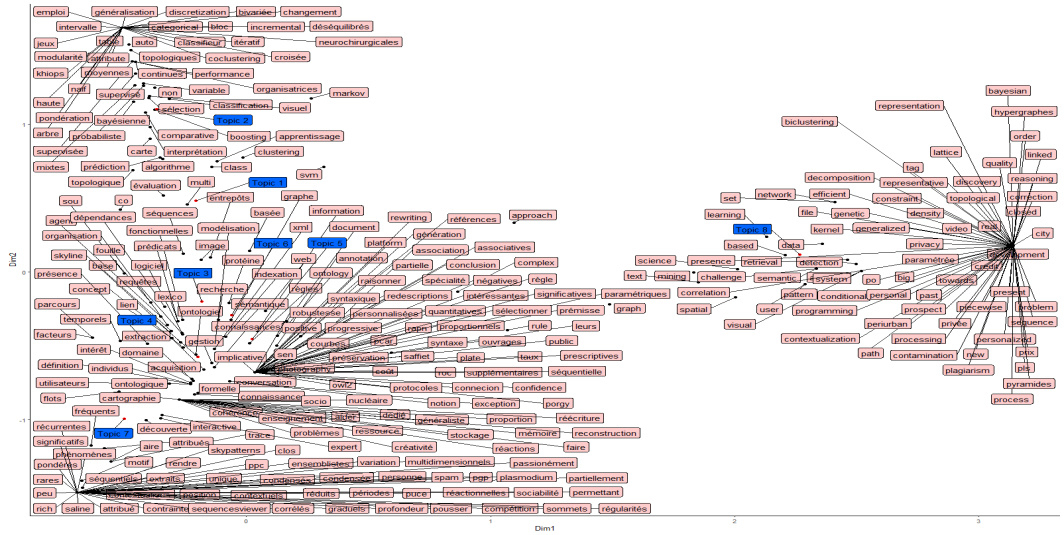


FIGURE 5.12: Topics description according to terms.

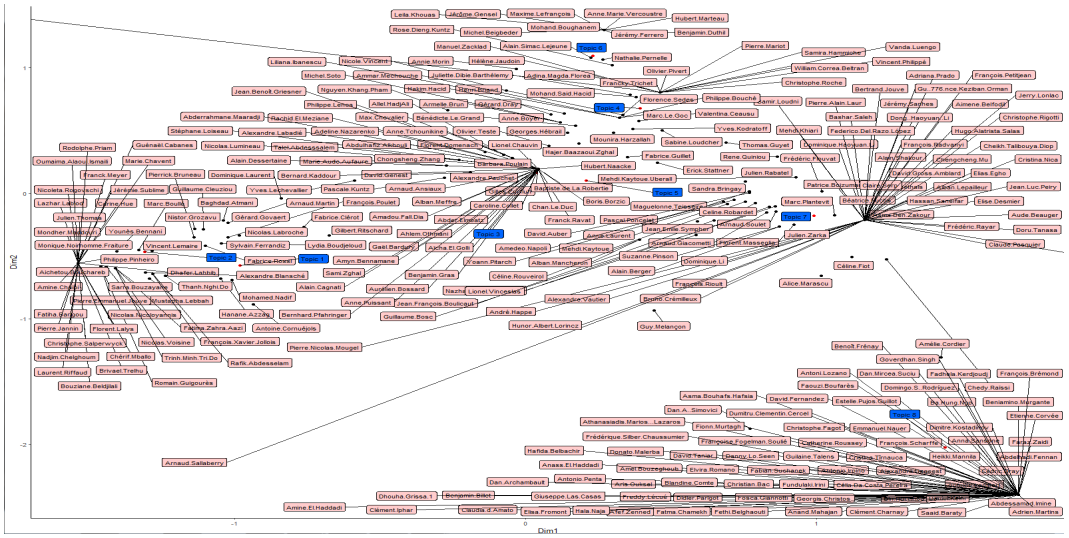


FIGURE 5.13: Description of topics according to authors.

How the topics evolve over time? We propose to study the evolution of the 8 previously extracted topics. For this, we calculate the number of papers according to topics and years. Figure 5.15 shows the evolution of the number of publications per topic and year. We have selected three topics namely topic 2, topic 4, and topic 8, which have different evolution behavior according to the time. We can notice that topic 2 does not follow a trend, but it is characterized by two peaks in 2008 and 2014, while topic 4 has a tendency decreasing. Finally, topic 8 represents English-language papers with a growing trend until 2016, which can be explained by an increase of non-Francophone researchers from 2001 to 2016.

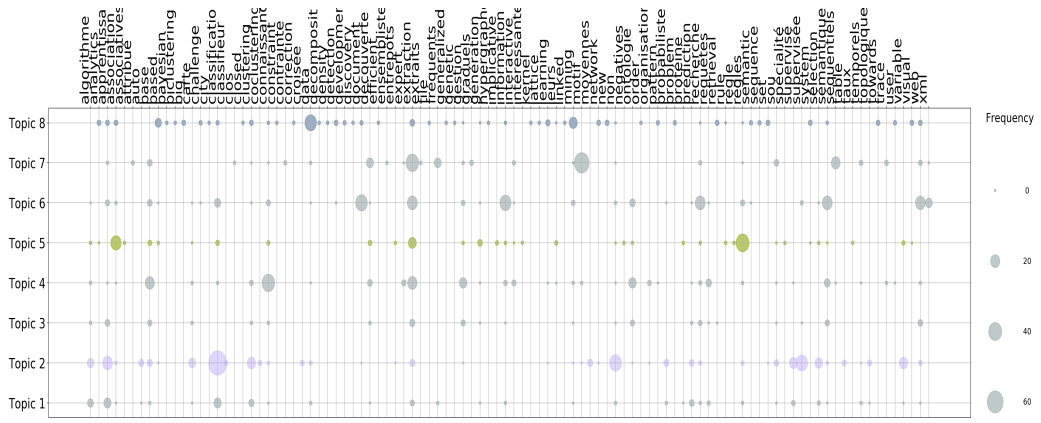


FIGURE 5.14: Frequencies of the most contributed terms to CA.

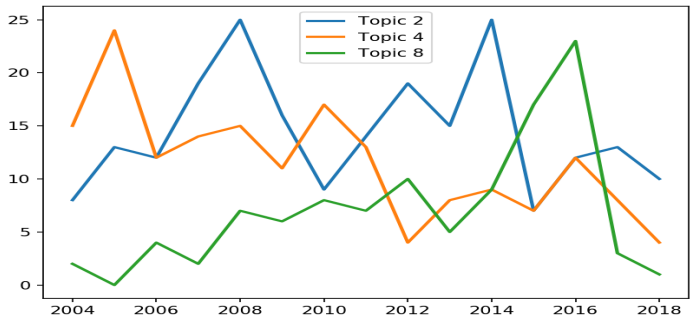


FIGURE 5.15: Evolution of topics over time.

5.2.3 Analysis of authors' communities

In order to analyze the author's communities, we have constructed several graphs representing different relationships between authors. We considered 816 authors for whom we were able to recover the affiliation. The constructed graphs are:

1. The co-terms matrix is constructed from the binarized authors-terms matrix and is computed by $\mathcal{B}\mathcal{B}^T$. It represents the number of terms commonly used by two authors.

2. The co-authors matrix is constructed from the documents-authors matrix by calculating $\mathcal{A}^T\mathcal{A}$. It represents the number of papers that two authors jointly drafted.

3. The co-affiliations matrix is constructed from the authors-affiliations matrix by calculating $\mathcal{H}\mathcal{H}^T$, where each cell equal to 1 indicates that two authors belong to the same institution, and 0 otherwise.

4. The co-topics matrix is constructed from the topics-authors matrix and computed by $\mathcal{G}\mathcal{G}^T$, where each cell equal to 1 indicates that two authors work on the same topics, and 0 otherwise.

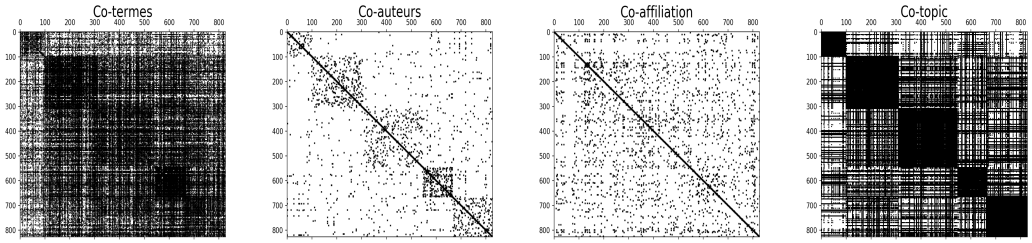


FIGURE 5.16: Reorganization of adjacency matrices of author's graphs.

Using these four relationships between authors, we construct the $Authors \times Authors \times Relationships$ tensor, with size $826 \times 826 \times 4$. Then, we applied the TSPLBM algorithm on this tenor using a number of clusters equals to 5 based on the modularity measure. The figure 5.16 represents the reorganization of the nodes of the four graphs using the partitioning obtained by TSPLBM.

How can interpret the communities of authors ? The proposed TSPLBM makes it possible to combine multiple information and thus simplify the interpretation of results. We built the topics-affiliations matrix and applied CA on this matrix. The figure 5.17 displays the results of CA, representing the different communities of authors as well as the affiliations that contribute the most. We can notice that community 1 mainly represents foreign researchers with domain names such as *@nac.ac.uk*, *@unicampania.it*, and *@uni-konstonz.de*.

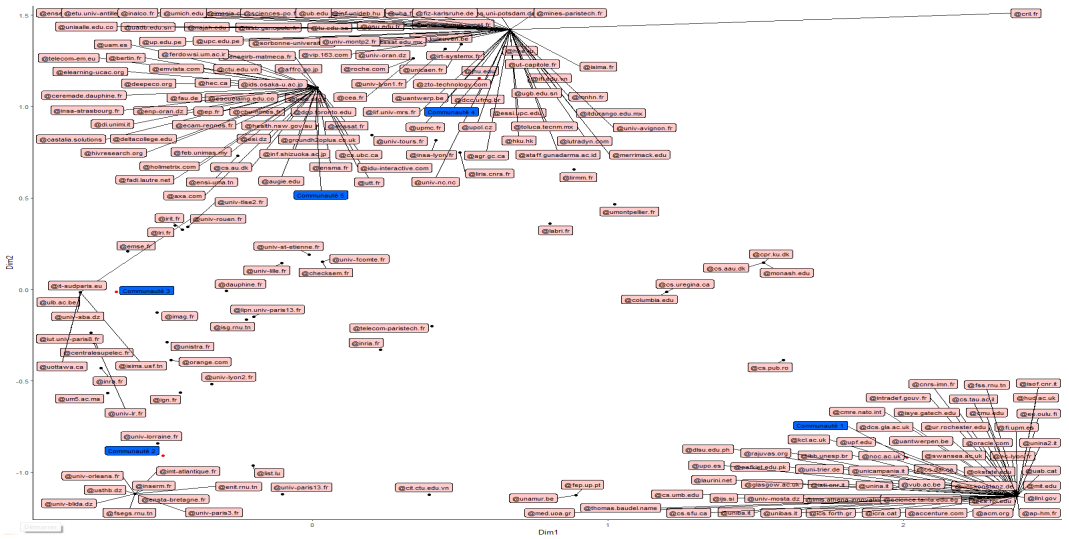


FIGURE 5.17: Description of author's communities by affiliation.

Are there communities of authors which stand out in terms of gender parity? We computed the male-female proportion for each community of authors. The aim is to appreciate the sex-ratio and the gender parity level in the different communities. Figure 5.18 shows the male-female proportion for the five authors' communities. It appears that all communities have almost the same proportion of men, 78-80%, and women 20-22%.

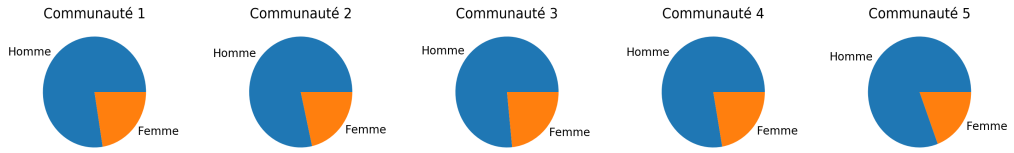


FIGURE 5.18: Male-female proportion by community of authors.

5.2.4 Recommendation of the *lecture committee*

We propose, in this part, a recommendation system to simplify the evaluation of submitted papers to the EGC conference. This system makes it possible to suggest researchers for reviewing the submitted papers. It is, therefore, simpler to set up a *Lecture committee* based on obtained recommendations. The papers used for recommendation are the papers that were not considered in the first part of topic modeling (section 5.2.2). Figure 5.19 represents the operating diagram of the proposed recommendation system. For a submitted paper to the EGC conference, the following steps are conducted:

1. Generate the vector representation of the title of the paper.
2. Construct a vector representation for each topic, from the documents-terms matrix (title) and the extracted topics.
3. Compute the cosine similarity between the vector representation of the title of the submitted paper and the vector representation of the topics.
4. Based on computed similarities, assign the new paper to one of the 8 topics.
5. Once the paper has been assigned to a topic, select the 30 authors who publish mostly in this topic, based on the EGC papers.
6. In order to improve the diversity and relevance of the recommendations, extract using the DBLP API, all published works available from the 30 selected authors.
7. Build a vector representation for each author based on the titles of all his publications available on DBLP.
8. Compute cosine similarity score between the title of the submitted paper and the authors' vector representation.
9. Recommend the three authors with the highest similarity scores to review the paper; under the condition that they do not belong to the same institution of one of the applicant authors.

We present in table 5.4 some examples of recommendation results. For instance, for the paper entitled "Scenario Ontology Analysis in a Big Data Context" the recommendation system proposes three reviewers which are *Fatiha Saïs*, *Stavrakas Yannis*, and *Thomas Tamisier* with scores of 0.293, 0.287, and 0.284, respectively. We notice that the ontologies and the semantic web are within the scope of authors and represent the topics of the paper.

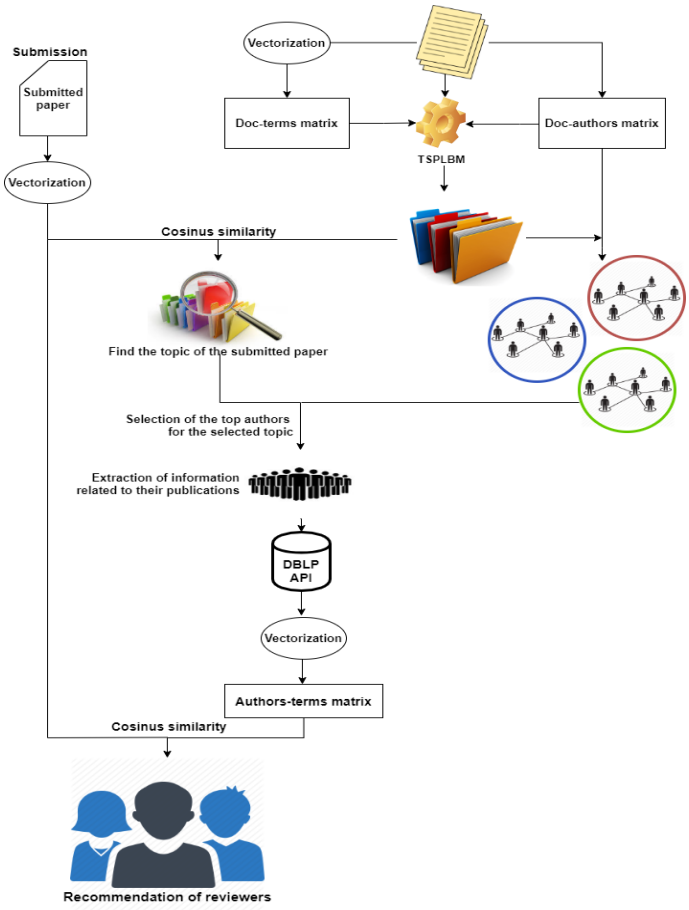


FIGURE 5.19: Recommendation system for reviewing research papers.

TABLE 5.4: Examples of obtained recommendations.

Titles	**Recommended reviewers**		
Analyse Ontologique de scénario dans un contexte	Fatiha Saïs	Stavrakas Yannis	Thomas Tamisier
Big Data	0.293	0.287	0.284
Big Data for understanding human dynamics	Stavrakas Yannis	Thomas Tamisier	Raja Chiky
the power of networks	0.331	0.328	0.317
Community structure in complex networks	Faraz Zaidi	Christine Largeron	Guy Melançon
	0.284	0.19	0.141
Détection de Singularités en temps-réel par combinaison	Alain Simac-Lejeune	Jérémy Ferrero	Thierry Despeyroux
d'apprentissage automatique et web sémantique basés sur Spark	0.269	0.194	0.174
eDOI : exploration itérative de grands graphes	David Genest	Djamel Abdelkader Zighed	Jean-Benoît Griesner
multi-couches basée sur une mesure de l'intérêt de l'utilisateur	0.125	0.119	0.093
Fouille de Motifs Graduels Fermés Fréquents	Lionel Vinceslas	Jean-Emile Symphor	François Rioult
Sous Contrainte de la Temporalité	0.156	0.124	0.105
Long-range influences in (social) networks	Nacéra Bennacer	Christine Largeron	Rim Faiz
	0.189	0.148	0.136
Méthode d'Apprentissage pour Extraire les Localisations	Emmanuel Viennet	Isabelle Tellier	Marc Boullé
dans les MicroBlogs	0.174	0.157	0.093

5.3 Conclusion

In this chapter, we presented tow challenging applications using tensor models. The first part was dedicated to the optimization of waste management. In fact, we applied the VEM−LBRM algorithm for the recommendation of collection's number and the waste containers' type. In the second part, we used TSPLBM for store clustering as the first step for the optimization of route collection. Finally, we used VEM−T for markdown analysis. The obtained results showed that our algorithms are efficient for these tasks.

In the second part, we applied TSPLBM to analyze the data of the EGC conference. Thus, tensor analysis of documents and authors allowed us to extract 8 topics and 5 authors' communities, respectively. We analyzed and described the topics using the extracted terms from titles, abstracts, and the authors whom contributed significantly. Similarly, authors' communities were analyzed using affiliations and exogenous variables such as the authors' *sex*. Finally, we presented a recommendation system of reviewers in order to compose the *lecture committee*. The obtained results highlight the relevance of the obtained recommendations.

Conclusion and Perspectives

Through this thesis, we have tackled the problem of three-way tensor co-clustering. We developed a class of models and algorithms tailored to the co-clustering of tensor data. The proposed algorithms are derived from the *latent block models* (LBM), which is suitable for different kinds of data, namely continuous and binary data, as well as contingency tables. Our focus on the above modeling assumption is motivated mainly by the adaptation of models to three-way tensor data that emerged from high dimensional datasets. We also focus on sparsity problem raising from the tensor structure using in several applications such as text-mining and recommender systems. Although various proposed tensor co-clustering techniques have been proven to be useful in this context, these latter are still severely challenged by the inherent characteristics of tensor data, namely the extreme sparsity, and high dimensionality. Thereby, the major contributions and results of this thesis can be summarized as follows:

- In **chapter 2**, inspired by the *Latent Block Model* (LBM), we proposed a novel *Tensor LBM* (or TLBM) designed from the ground up to deal with three-way data, instead of relying on factorization approaches, which main focus is not clustering. Our study showed that the proposed TLBM allows handling a three-way tensor effectively, considering different kinds of data (continuous, binary, and contingency tables) referred to as. The TLBM model is beneficial from several perspectives: it is parsimonious, allows us to make precise assumptions, and gives rise to various co-clustering algorithms, including hard and soft variants. Our proposal is both straightforward and more effective than a variety of other clustering and co-clustering techniques devoted to the same tasks. It proved its effectiveness in several applications, namely the recommender system, hyperspectral image clustering, and document categorization.

- In **chapter 3**, we described a novel probabilistic model, denoted as *Tensor Sparse Poisson Latent Block Model* (TSPLBM). TSPLBM is based on SPLBM and designed to deal with the sparsity and high dimensionality of tensor data. Further, it is parsimonious, leads to effective co-clustering of multiple graphs, and can be viewed as an implicit consensus graph clustering. In addition, to evaluate the performance of the proposed `TSPLBM` algorithm in terms of consensus, we provided a detailed comparison with traditional consensus clustering approaches, This experimentation reveals the advantages of the implicit consensus obtained by `TSPLBM` comparing to traditional consensus.

- In **chapter 4**, In this chapter, we proposed a novel model for co-clustering and prediction, which can be viewed as a co-clusterwise model combining simultaneously unsupervised and supervised learning. As we consider the challenge of recommender systems, the proposed model refereed to as *Latent Block Regression Model* (LBRM) is based on the realistic assumption that users who have the same profiles (age category,

sex, occupation, etc.) tend to share similar tastes. Thereby, in this spirit, we proposed the VEM-LBRM algorithm, which simultaneously seeks for groups of users having similar profiles, and items with similar characteristics to predict ratings. On synthetic and real-world datasets, that include both covariates (of users and items), the proposed algorithm demonstrates its advantages and the interest of including information about users and items.

In previous chapters, we have evaluated all of the proposed algorithms with some datasets commonly used in the unsupervised learning community to study the performances of a clustering method. **Chapter 5** has been devoted to real applications. First, we have shown the usefulness of the proposed models in the waste management field, where a lot of information about customers and waste production are available. Thereby, we offered a recommendation system using the proposed VEM-LBRM co-clustering algorithm. Moreover, we used TSPLBM for store clustering as the first step for the optimization of waste collection routes. Also, we used the VEM-T algorithm for markdown analysis. The obtained results showed the effectiveness of our proposal for the optimization of waste management. Secondly, we have applied TSPLBM in the context of the *EGC Conference Challenge*. The derived algorithms were successfully used to discover topics from published papers and also authors' communities. These results lead to developing a recommender system for composing the *lecture committee*. The obtained results enable a better understanding of the dynamic of the *EGC conference*.

The studies presented in this thesis motivate further issues that we intend to investigate:

- In the model selection context, the criteria such as AIC [Akaike, 1998], BIC [Schwarz, 1978], or ICL [Biernacki et al., 2000] can be adapted. An extension of some researches performed in co-clustering ([Vu and Aitkin, 2015]) should be interesting to this end. Noting that assessing the number of co-clusters is no considered in this thesis and could be dealt with these criteria.

- With the three-way tensor datasets considered in the thesis, the dimension of the third mode is not high (between 3 and 42 slices). In certain situations, this dimension can be higher, and a tri-clustering could be more beneficial than co-clustering to extract relevant tri-clusters.

- In the mixture model, it is easy to show that some classical dissimilarity measures are associated to probability distributions. Hence, it would be interesting to study possible connections between the tensor factorization methods and TLBM.

Appendix A

Updating of Common Parameters of Tensor Models

A.1 Update $\tilde{z}_{ik}$ and $\tilde{w}_{j\ell}$ $\forall i,k,j,\ell$ for TLBM

To obtain the expression of $\tilde{z}_{ik}$, we maximize the above soft criterion $F_C(\tilde{\mathbf{z}},\tilde{\mathbf{w}};\Omega)$ with respect to $\tilde{z}_{ik}$, subject to the constraint $\sum_k \tilde{z}_{ik}=1$. The corresponding Lagrangian, up to terms which are not function of $\tilde{z}_{ik}$, is given by :

$$\begin{aligned} L(\tilde{\mathbf{z}},\beta) = &\sum_{i,k}\tilde{z}_{ik}\log\pi_k + \sum_{i,j,k,\ell}\tilde{z}_{ik}\tilde{w}_{jk}\log(\Phi(\mathbf{x}_{ij},\boldsymbol{\lambda}_{k\ell})) \\ &-\sum_{i,k}\tilde{z}_{ik}\log(\tilde{z}_{ik}) + \beta(1-\sum_k \tilde{z}_{ik}). \end{aligned} \tag{A.1}$$

Where $\Phi(\mathbf{x}_{ij},\boldsymbol{\lambda}_{k\ell})$ depends on the probability distribution. Taking derivatives with respect to $\tilde{z}_{ik}$, we obtain:

$$\frac{\partial L(\tilde{\mathbf{z}},\beta)}{\partial \tilde{z}_{ik}} = \log\pi_k + \sum_{j,\ell} w_{j\ell}\log(\Phi(\mathbf{x}_{ij},\boldsymbol{\lambda}_{k\ell})) - \log\tilde{z}_{ik} - 1 - \beta.$$

Setting this derivative to zero yields:

$$\tilde{z}_{ik} = \frac{\pi_k \exp(\sum_{j,\ell} w_{j\ell}\log(\Phi(\mathbf{x}_{ij},\boldsymbol{\lambda}_{k\ell}))}{\exp(\beta+1)}.$$

Summing both sides over all k' yields

$$\exp(\beta+1) = \sum_{k'}\pi_{k'}\exp(\sum_{j,\ell}\tilde{w}_{j\ell}\log(\Phi(\mathbf{x}_{ij},\boldsymbol{\lambda}_{k'\ell})).$$

Plugging $\exp(\beta)$ in $\tilde{z}_{ik}$ leads to:

$$\tilde{z}_{ik} \propto \pi_k \exp(\sum_{j,\ell}\tilde{w}_{j\ell}\log(\Phi(\mathbf{x}_{ij},\boldsymbol{\lambda}_{k\ell})).$$

In the same way, we can estimate $\tilde{w}_{jl}$ maximizing $F_C(\tilde{\mathbf{z}}, \tilde{\mathbf{w}}; \Omega)$ with respect to $\tilde{w}_{j\ell}$, subject to the constraint $\sum_\ell \tilde{w}_{j\ell} = 1$; we obtain

$$\tilde{w}_{j\ell} \propto \rho_k \exp(\sum_{i,k} \tilde{z}_{ik} \log(\Phi(\mathbf{x}_{ij}, \boldsymbol{\lambda}_{k\ell})).$$

A.2 Estimation of the π_k and ρ_ℓ $\forall k, \ell$ of TLBM

Derivation of row proportion π_k

Taking into account the constaint $\sum_k \pi_k = 1$

$$\frac{\partial L_C}{\partial \pi_k} = \frac{\partial \left\{ \sum_i \sum_k \tilde{z}_{ik} \log(\pi_k) + \lambda \left(1 - \sum_k \pi_k\right) \right\}}{\partial \pi_k} \tag{A.2}$$

$$\frac{\partial L_C}{\partial \pi_k} = \frac{\sum_i \tilde{z}_{ik}}{\pi k} - \lambda \tag{A.3}$$

Setting the partial derivative to zero, we obtain for each k: $\lambda = \frac{\sum_i \tilde{z}_{ik}}{\pi k}$
For this to be true, π_k must be proportional to $\sum_i \tilde{z}_{ik}$, then :

$$\pi_k = \frac{\sum_i \tilde{z}_{ik}}{\sum_{k'} \sum_i \tilde{z}_{ik}} \tag{A.4}$$

Finally:

$$\pi_k = \frac{\sum_i \tilde{z}_{ik}}{n} \tag{A.5}$$

Derivation of column proportion ρ_ℓ

Taking into account the constaint $\sum_\ell \rho_\ell = 1$

$$\frac{\partial L_C}{\partial \rho_\ell} = \frac{\partial \left\{ \sum_j \sum_\ell \tilde{w}_{j\ell} \log(\rho_\ell) + \lambda \left(1 - \sum_\ell \rho_\ell\right) \right\}}{\partial \rho_\ell} \tag{A.6}$$

$$\frac{\partial L_C}{\partial \rho_\ell} = \frac{\sum_j \tilde{w}_{j\ell}}{\rho_\ell} - \lambda \tag{A.7}$$

Setting the partial derivative to zero, we obtain for each ℓ: $\lambda = \frac{\sum_j \tilde{w}_{j\ell}}{\rho \ell}$
For this to be true, ρ_ℓ must be proportional to $\sum_j \tilde{w}_{j\ell}$, then :

$$\rho_\ell = \frac{\sum_j \tilde{w}_{j\ell}}{\sum_{\ell'} \sum_j \tilde{w}_{j\ell}} \tag{A.8}$$

$$\rho_\ell = \sum_j \frac{\tilde{w}_{j\ell}}{d} \tag{A.9}$$

Appendix B

Estimation of TLBM's Parameters

For more clarity, we can decompose the TLBM log-likelihood function as follows:

$$L_C = L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega}) = \sum_{j\ell} \tilde{w}_{j\ell} \log(\rho_\ell) + \sum_{ik} \tilde{z}_{ik} \log(\pi_k) + \sum_{k\ell} \mathcal{L}_C^{k\ell}$$

where $\mathcal{L}_C^{k\ell}$ depends on probability distribution function per block. Howsever, it can be, also, depends on slice b of the tensor. The log-likelihood can be expressed as follows:

$$L_C = L_C(\tilde{\mathbf{Z}}, \tilde{\mathbf{W}}, \boldsymbol{\Omega}) = \sum_{j\ell} \tilde{w}_{j\ell} \log(\rho_\ell) + \sum_{ik} \tilde{z}_{ik} \log(\pi_k) + \sum_{k\ell} \sum_{b} \mathcal{L}_C^{k\ell^b}$$

B.1 Estimation of the $\mu_{k\ell}$ and $\Sigma_{k\ell}$ $\forall k, \ell$ parameters of Gaussian TLBM

Considering a Gaussian TLBM, the $\mu_{k\ell}$ and $\Sigma_{k\ell}$ parameters can be obtained from the following derivatives:

$$\frac{\partial \mathcal{L}_C^{k\ell}}{\partial \boldsymbol{\mu}_{k\ell}} \quad \text{and} \quad \frac{\partial \mathcal{L}_C^{k\ell}}{\partial \boldsymbol{\Sigma}_{k\ell}}$$

where

$$\mathcal{L}_C^{k\ell} = -\frac{1}{2} \tilde{z}_{.k} \tilde{w}_{.\ell} \log |\Sigma_{k\ell}| - \frac{1}{2} \sum_{i,j} \tilde{z}_{ik} \tilde{w}_{j\ell} (\mathbf{x}_{ij} - \boldsymbol{\mu}_{k\ell})^\top \boldsymbol{\Sigma}_{k\ell}^{-1} (\mathbf{x}_{ij} - \boldsymbol{\mu}_{k\ell}),$$

with $\tilde{z}_{.k} = \sum_i \tilde{z}_{ik}$ and $\tilde{w}_{.\ell} = \sum_j \tilde{w}_{j\ell}$. The following formulas involving the vector-by-vector ($\mathbf{x}$) and matrix-by-matrix ($\mathbf{M}$) derivates.

$$\begin{cases} \frac{\partial \mathbf{x}^\top \mathbf{M} \mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{M}\mathbf{x}, \\ \frac{\partial \log |\mathbf{M}|}{\partial \mathbf{M}} = (\mathbf{M}^{-1})^\top, \\ \frac{\partial \mathbf{x}^\top \mathbf{M} \mathbf{x}}{\partial \mathbf{M}} = (\mathbf{M}^{-1}) \mathbf{x}\mathbf{x}^\top (\mathbf{M}^{-1})^\top \end{cases}$$

lead to

$$\frac{\partial \mathcal{L}_C^{k\ell}}{\partial \boldsymbol{\mu}_{k\ell}} = -\sum_{i,j} \tilde{z}_{ik} \tilde{w}_{j\ell} \boldsymbol{\Sigma}_{k\ell}^{-1} (\mathbf{x}_{ij} - \boldsymbol{\mu}_{k\ell})$$

and

$$\frac{\partial \mathcal{L}_C^{k\ell}}{\partial \boldsymbol{\Sigma}_{k\ell}} = -\tilde{z}_{.k}\tilde{w}_{.\ell}\log(\Sigma_{k\ell}^{-1})^\top + \frac{1}{2}\sum_{i,j}\tilde{z}_{ik}\tilde{w}_{j\ell}(\boldsymbol{\Sigma}_{k\ell}^{-1})^\top(\mathbf{x}_{ij}-\boldsymbol{\mu}_{k\ell})(\mathbf{x}_{ij}-\boldsymbol{\mu}_{k\ell})^\top(\boldsymbol{\Sigma}_{k\ell}^{-1})^\top.$$

The two partial derivatives set to 0 lead to

$$\hat{\boldsymbol{\mu}}_{k\ell} = \frac{\sum_{i,j}\tilde{z}_{ik}\tilde{w}_{j\ell}\mathbf{x}_{ij}}{\sum_{i,j}\tilde{z}_{ik}\tilde{w}_{j\ell}},$$

and

$$\hat{\boldsymbol{\Sigma}}_{k\ell} = \frac{\sum_{i,j}\tilde{z}_{ik}\tilde{w}_{j\ell}(\mathbf{x}_{ij}-\boldsymbol{\mu}_{k\ell})(\mathbf{x}_{ij}-\boldsymbol{\mu}_{k\ell})^\top}{\sum_{i,j}\tilde{z}_{ik}\tilde{w}_{j\ell}}.$$

B.2 Estimation of the $\boldsymbol{\mu}_{k\ell}^b$'s of Bernoulli TLBM

Considering a Bernoulli TLBM, the $\mu_{k\ell}^b$ parameter can be obtained from the following derivatives:

$$\frac{\partial \mathcal{L}_C^{k\ell^b}}{\partial \mu_{k\ell}^b} = \frac{\partial \sum_{i,j} z_{ik}w_{j\ell}\left(\log(1-\mu_{k\ell}^b) + x_{ij}^b \log\frac{\mu_{k\ell}^b}{1-\mu_{k\ell}^b}\right)}{\partial \mu_{k\ell}^b}$$

Considering only the terms which depend on $\mu_{k\ell}^b$:

$$\frac{\partial \mathcal{L}_C^{k\ell^b}}{\partial \mu_{k\ell}^b} = \frac{\partial \sum_{i,j} z_{ik}w_{j\ell}\left(\log(1-\mu_{k\ell}^b) + x_{ij}^b\left(\log(\mu_{k\ell}^b) - \log(1-\mu_{k\ell}^b)\right)\right)}{\partial \mu_{k\ell}^b}$$

Applying the derivative and using logarithm propriety derivation $\log(x) = \frac{1}{x}$, we obtain:

$$\sum_{i,j} z_{ik}w_{j\ell}\left(-\frac{1}{1-\mu_{k\ell}^b} + \frac{x_{ij}^b}{\mu_{k\ell}^b} + \frac{x_{ij}^b}{1-\mu_{k\ell}^b}\right) = \sum_{i,j} z_{ik}w_{j\ell}\left(\frac{x_{ij}^b}{\mu_{k\ell}^b} - \frac{1-x_{ij}^b}{1-\mu_{k\ell}^b}\right) = 0.$$

After some simplifications:

$$(1-\mu_{k\ell}^b)\sum_{i,j} z_{ik}w_{j\ell}x_{ij}^b = \mu_{k\ell}^b \sum_{i,j} z_{ik}w_{j\ell}(1-x_{ij}^b)$$

Finally, we obtain the following expression of $\boldsymbol{\mu}_{k\ell}^b$:

$$\hat{\boldsymbol{\mu}}_{k\ell} = \frac{\sum_{i,j}\tilde{z}_{ik}\tilde{w}_{j\ell}\mathbf{x}_{ij}}{\sum_{i,j}\tilde{z}_{ik}\tilde{w}_{j\ell}}.$$

B.3 Estimation of the $\gamma_{k\ell}^b$'s of Poisson TLBM

Considering a Poisson TLBM, the $\gamma_{k\ell}^b$ parameter can be obtained from the following derivatives:

$$\frac{\partial \mathcal{L}_C^{k\ell^b}}{\partial \gamma_{k\ell}^b} = \frac{\partial \sum_{i,j} z_{ik} w_{j\ell} \left(-x_{i.}^b x_{.j}^b \gamma_{k\ell}^b + x_{ij}^b \log(\gamma_{k\ell}^b) \right) \}}{\partial \gamma_{k\ell}^b}$$

Applying the derivative and using logarithm propriety derivation $\log(x) = \frac{1}{x}$:

$$\sum_{i,j} z_{ik} w_{j\ell} \left(-x_{i.}^b x_{.j}^b + x_{ij}^b \frac{1}{\gamma_{k\ell}^b} \right) = 0$$

Using the distributive property of product operator:

$$\sum_{i,j} z_{ik} w_{j\ell} x_{ij}^b \frac{1}{\gamma_{k\ell}^b} = \sum_{i,j} z_{ik} w_{j\ell} x_{i.}^b x_{.j}^b$$

And,

$$\sum_{i,j} z_{ik} w_{j\ell} x_{i.}^b x_{.j}^b = \sum_i z_{ik} x_{i.}^b \sum_j w_{j\ell} x_{.j}^b$$

Finally, we obtain :

$$\gamma_{k\ell}^b = \frac{\sum_{i,j} z_{ik} w_{j\ell} x_{ij}^b}{\sum_i z_{ik} x_{i.}^b \sum_j w_{j\ell} x_{.j}^b}$$

Appendix C

Estimation of TSPLBM's Parameters

C.1 Estimation of the γ_{kk}^{b} parameter

$$\frac{\partial \mathcal{L}_C^{k\ell^b}}{\partial \gamma_{kk}^b} = \frac{\partial\{\left[x_{kk}^b \log(\frac{\gamma_{kk}^b}{\gamma^b}) - x_{k.}^b x_{.k}^b(\gamma_{kk}^b - \gamma^b)\right] + N_b(\log(\gamma) - N_b\gamma)\}}{\partial \gamma_{kk}^b}$$

Considering only the terms which depend on γ_{kk}^b:

$$\frac{\partial\{x_{kk}^b \log(\frac{\gamma_{kk}^b}{\gamma^b}) - x_{k.}^b x_{.k}^b(\gamma_{kk}^b - \gamma^b)\}}{\partial \gamma_{kk}^b} = \frac{\partial\{\left(x_{kk}^b(\log(\gamma_{kk}^b) - \log(\gamma^b)) - x_{k.}^b x_{.k}^b(\gamma_{kk}^b - \gamma^b)\right)\}}{\partial \gamma_{kk}^b}$$

Applying the derivative and using logarithm propriety derivation, we obtain:

$$x_{kk}^b \frac{1}{\gamma_{kk}^b} - x_{k.}^b x_{.k}^b = 0$$

Then,

$$\gamma_{kk}^b = \frac{x_{kk}^b}{x_{k.}^b x_{.k}^b}$$

C.2 Estimation of the γ^b parameter

$$\frac{\partial \mathcal{L}_C^{k\ell^b}}{\partial \gamma^b} = \frac{\partial\{\left[x_{kk}^b \log(\frac{\gamma_{kk}^b}{\gamma^b}) - x_{k.}^b x_{.k}^b(\gamma_{kk}^b - \gamma^b)\right] + N_b(\log(\gamma) - N_b\gamma)\}}{\partial \gamma^b}$$

Considering only the terms which depend on γ^b:

$$\frac{\partial \mathcal{L}_C^{k\ell^b}}{\partial \gamma^b} = \frac{\partial\{\left(x_{kk}^b(\log(\gamma_{kk}^b) - \log(\gamma^b)) - x_{k.}^b x_{.k}^b(\gamma_{kk}^b - \gamma^b)\right) + N_b(\log(\gamma) - N_b\gamma)\}}{\partial \gamma^b}$$

Applying the derivative and using logarithm propriety derivation, we obtain:

$$\frac{-x_{kk}^b}{\gamma^b} - x_{k.}^b x_{.k}^b + \frac{N_b}{\gamma^b} - N_b^2 = \frac{N_b - x_{kk}^b}{\gamma^b} + x_{k.}^b x_{.k}^b - N_b^2 = 0$$

Finally, we obtain:

$$\gamma^b = \frac{N_b - x_{kk}^b}{N_b^2 - x_{k.}^b x_{.k}^b}$$

Appendix D

Estimation of LBRM's Parameters

D.1 Estimation of the $\beta_{k\ell}$ parameter

Considering a Poisson TLBM, the $\beta_{k\ell}$ parameter can be obtained from the following derivatives:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathrm{C}}^{k\ell}}{\partial \boldsymbol{\beta}_{k\ell}'} &= \frac{\partial\{\sum_{i,j} \frac{\tilde{z}_{ik}\tilde{w}_{j\ell}}{2}\left(\log({\sigma_{k\ell}^2}^{-1}) - \frac{(y_{ij}-\boldsymbol{\beta}_{k\ell}'\mathbf{x}_{ij})^2}{\sigma_{k\ell}^2}\right)\}}{\partial \boldsymbol{\beta}_{k\ell}} \\ &= \frac{\partial\left\{\sum_{i,j} \frac{-\tilde{z}_{ik}\tilde{w}_{j\ell}}{2}\left[(y_{ij}-\boldsymbol{\beta}_{k\ell}'\mathbf{x}_{ij})'{\sigma_{k\ell}^2}^{-1}(y_{ij}-\boldsymbol{\beta}_{k\ell}'\mathbf{x}_{ij})\right]\right\}}{\partial \boldsymbol{\beta}_{k\ell}'} \\ &= \frac{\partial\{\sum_{i,j} \frac{-\tilde{z}_{ik}\tilde{w}_{j\ell}}{2}\left[-y_{ij}'{\sigma_{k\ell}^2}^{-1}\boldsymbol{\beta}_{k\ell}'\mathbf{x}_{ij} - \mathbf{x}_{ij}'\boldsymbol{\beta}_{k\ell}{\sigma_{k\ell}^2}^{-1}y_{ij} + \mathbf{x}_{ij}'\boldsymbol{\beta}_{k\ell}{\sigma_{k\ell}^2}^{-1}\boldsymbol{\beta}_{k\ell}'\mathbf{x}_{ij}\right]\}}{\partial \boldsymbol{\beta}_{k\ell}'}\end{aligned}$$

Using trace properties:

$$\frac{\partial \mathcal{L}_{\mathrm{C}}^{k\ell}}{\partial \boldsymbol{\beta}_{k\ell}'} = \frac{\partial\{\sum_{i,j} \frac{\tilde{z}_{ik}\tilde{w}_{j\ell}}{2}\left[tr(\boldsymbol{\beta}_{k\ell}'\mathbf{x}_{ij}y_{ij}'{\sigma_{k\ell}^2}^{-1}) + tr({\sigma_{k\ell}^2}^{-1}y_{ij}\mathbf{x}_{ij}'\boldsymbol{\beta}_{k\ell}') - \boldsymbol{\beta}_{k\ell}'\mathbf{x}_{ij}\mathbf{x}_{ij}'\boldsymbol{\beta}_{k\ell}{\sigma_{k\ell}^2}^{-1}\right]\}}{\partial \boldsymbol{\beta}_{k\ell}'}$$

Applying the derivative:

$$\sum_{i,j} \frac{\tilde{z}_{ik}\tilde{w}_{j\ell}}{2}\left[{\sigma_{k\ell}^2}^{-1}y_{ij}\mathbf{x}_{ij}' + {\sigma_{k\ell}^2}^{-1}y_{ij}\mathbf{x}_{ij}' - \left({\sigma_{k\ell}^2}^{-1}\boldsymbol{\beta}_{k\ell}'\mathbf{x}_{ij}\mathbf{x}_{ij}' + {\sigma_{k\ell}^2}^{-1}\boldsymbol{\beta}_{k\ell}'\mathbf{x}_{ij}\mathbf{x}_{ij}'\right)\right] = 0$$

Finally:

$$\boldsymbol{\beta}_{k\ell} = \left(\sum_{i,j} \tilde{z}_{ik}w_{j\ell}y_{ij}\mathbf{x}_{ij}'\right)\left(\sum_{i,j} \tilde{z}_{ik}w_{j\ell}\mathbf{x}_{ij}\mathbf{x}_{ij}'\right)^{-1}$$

D.2 Estimation of the $\sigma^2_{k\ell}$ parameter

Considering a Poisson TLBM, the $\sigma^2_{k\ell}$ parameter can be obtained from the following derivatives:

$$\frac{\partial \mathcal{L}_C^{k\ell}}{\partial {\sigma^2_{k\ell}}^{-1}} = \frac{\partial \{\sum_{i,j} \frac{\tilde{z}_{ik}\tilde{w}_{j\ell}}{2}\left(\log({\sigma^2_{k\ell}}^{-1}) - \frac{(y_{ij}-\boldsymbol{\beta}'_{k\ell}\mathbf{x}_{ij})^2}{\sigma^2_{k\ell}}\right)\}}{\partial {\sigma^2_{k\ell}}^{-1}}$$

Using logarithm properties: $-\log(x) = \log(x^{-1})$, we obtain:

$$\frac{\partial \mathcal{L}_C^{k\ell}}{\partial {\sigma^2_{k\ell}}^{-1}} = \frac{\partial \{\sum_{i,j} \frac{1}{2}\tilde{z}_{ik}w_{j\ell}\left(\log({\sigma^2_{k\ell}}^{-1}) - {\sigma^2_{k\ell}}^{-1}(y_{ij}-\boldsymbol{\beta}'_{k\ell}\mathbf{x}_{ij})^2\right)\}}{\partial {\sigma^2_{k\ell}}^{-1}}$$

Taking the derivative:

$$\frac{1}{2}\sum_{i,j}\tilde{z}_{ik}w_{j\ell}\left\{({\sigma^2_{k\ell}}^{-1})^{-1} - (y_{ij}-\boldsymbol{\beta}'_{k\ell}\mathbf{x}_{ij})^2\right\} = 0$$

$$\sum_{i,j}\tilde{z}_{ik}w_{j\ell}(\sigma^2_{k\ell}) - \sum_{i,j}\tilde{z}_{ik}w_{j\ell}(y_{ij}-\boldsymbol{\beta}'_{k\ell}\mathbf{x}_{ij})^2 = 0$$

$$\sigma^2_{k\ell} = \frac{\sum_{i,j}\tilde{z}_{ik}w_{j\ell}(y_{ij}-\boldsymbol{\beta}'_{k\ell}\mathbf{x}_{ij})^2}{\sum_{i,j}z_{ik}w_{j\ell}}$$

Bibliography

E. E. Abdallah, A. B. Hamza, and P. Bhattacharya. Mpeg video watermarking using tensor singular value decomposition. In M. Kamel and A. Campilho, editors, *Image Analysis and Recognition*, pages 772–783, 2007. ISBN 978-3-540-74260-9. doi: 10.1007/978-3-540-74260-9_69. URL https://link.springer.com/chapter/10.1007/978-3-540-74260-9_69.

D. Agarwal and S. Merugu. Predictive discrete latent factor models for large scale dyadic data. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 26–35, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936097. doi: 10.1145/1281192.1281199. URL https://doi.org/10.1145/1281192.1281199.

H. A. Ahmed, P. Mahanta, D. K. Bhattacharyya, J. K. Kalita, and A. Ghosh. Intersected coexpressed subcube miner: An effective triclustering algorithm. In *2011 World Congress on Information and Communication Technologies*, pages 846–851, Dec 2011. doi: 10.1109/WICT.2011.6141358. URL https://ieeexplore.ieee.org/document/6141358.

M. Ailem, F. Role, and M. Nadif. Co-clustering document-term matrices by direct maximization of graph modularity. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, page 1807–1810, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450337946. doi: 10.1145/2806416.2806639. URL https://doi.org/10.1145/2806416.2806639.

M. Ailem, F. Role, and M. Nadif. Model-based co-clustering for the effective handling of sparse data. *Pattern Recognition*, 72:108 – 122, 2017. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2017.06.005. URL http://www.sciencedirect.com/science/article/pii/S0031320317302297.

M. Ailem, F. Role, and M. Nadif. Sparse poisson latent block model for document clustering. *IEEE Transactions on Knowledge and Data Engineering*, 29(7):1563–1576, 2017. ISSN 2326-3865. doi: 10.1109/TKDE.2017.2681669. URL https://ieeexplore.ieee.org/abstract/document/7876732.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, 1998.

K. Allab, L. Labiod, and M. Nadif. Seminmf-pca framework for sparse data co-clustering. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 347–356, 2016.

K. Allab, L. Labiod, and M. Nadif. Multi-manifold matrix decomposition for data co-clustering. *Pattern Recognition*, 64:386–398, 2017.

T. Alqurashi and W. Wang. Clustering ensemble method. *International Journal of Machine Learning and Cybernetics*, 10(6):1227–1246, Jun 2019. doi: 10.1007/s13042-017-0756-7. URL https://link.springer.com/article/10.1007/s13042-017-0756-7.

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. volume 28, pages 49–60, New York, NY, USA, 1999. Association for Computing Machinery. doi: 10.1145/304181.304187. URL https://doi.org/10.1145/304181.304187.

B. W. Bader and T. G. Kolda. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Trans. Math. Softw.*, 32(4):635–653, 2006. ISSN 0098-3500. doi: 10.1145/1186785.1186794. URL https://doi.org/10.1145/1186785.1186794.

A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. J. Mooney. Model-based overlapping clustering. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, page 532–537. Association for Computing Machinery, 2005. ISBN 159593135X. doi: 10.1145/1081870.1081932. URL https://doi.org/10.1145/1081870.1081932.

A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In *SIAM international conference on data mining*, pages 145–156, 2007. doi: 10.1137/1.9781611972771.14. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972771.14.

E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139, 1999. doi: 10.1023/A:1007515423169. URL https://link.springer.com/article/10.1023/A:1007515423169.

J.-P. Benzecri. *L'analyse des données, tome 2 : l'analyse des correspondances*. Dunod, Paris, 1973.

C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7):719–725, 2000.

V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008 (10):P10008, 2008. doi: 10.1088/1742-5468/2008/10/p10008. URL https://doi.org/10.1088%2F1742-5468%2F2008%2F10%2Fp10008.

H. Bock. Simultaneous clustering of objects and variables. *E. Diday (ed) Analyse des données et Informatique*, page 187–203, 1979.

H.-H. Bock. *Clustering Methods: A History of k-Means Algorithms*, pages 161–172. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-73560-1. doi:

10.1007/978-3-540-73560-1_15. URL https://doi.org/10.1007/978-3-540-73560-1_15.

F. Bourgeois and J.-C. Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Commun. ACM*, 14(12):802–804, Dec. 1971. ISSN 0001-0782. doi: 10.1145/362919.362945. URL https://doi.org/10.1145/362919.362945.

D. Bourguignon. Turning waste into a resource moving towards a 'circular economy'. *European Parliament: Briefing*, 2014. URL https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS_BRI(2014)545704.

D. Bourguignon. Understanding waste management policy challenges and opportunities. *European Parliament: Briefing*, 2015. URL https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS_BRI(2015)559493.

R. Boutalbi, L. Labiod, and M. Nadif. Classification croisée et regression locale. In *SFC conference (Société Francophone de Classification)*, pages 87–90, 2018.

R. Boutalbi, L. Labiod, and M. Nadif. Co-clustering from tensor data. In *Advances in Knowledge Discovery and Data Mining*, pages 370–383. Springer International Publishing, 2019a. ISBN 978-3-030-16148-4. doi: 10.1007/978-3-030-16148-4_29. URL https://link.springer.com/chapter/10.1007/978-3-030-16148-4_29.

R. Boutalbi, L. Labiod, and M. Nadif. 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 2019, Paris, France, July 21-25, 2019*, pages 1157–1160. ACM, 2019b.

R. Boutalbi, L. Labiod, and M. Nadif. Implicit consensus clustering from multiple graphs. 2020.

X. Cao, X. Wei, Y. Han, and D. Lin. Robust face clustering via tensor decomposition. *IEEE Transactions on Cybernetics*, 45(11):2546–2557, 2015. ISSN 2168-2275. doi: 10.1109/TCYB.2014.2376938.

J. D. Carroll and C. J. J. Analysis of individual differences in multidimensional scaling via an n-way generalizarion of "eckart-young" decomposition. *Psychometrika*, 35: 283–319, 1970. doi: 10.1007/BF02310791. URL https://link.springer.com/article/10.1007/BF02310791.

J. D. Carroll, S. Pruzansky, and K. J. B. Candelinc: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika*, 45:3–24, 1980. doi: 10.1007/BF02293596. URL https://link.springer.com/article/10.1007/BF02293596.

G. Celeux and G. Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14(3):315–332, 1992. doi: 10.1016/0167-9473(92)90042-E. URL https://www.sciencedirect.com/science/article/pii/016794739290042E.

C. Chen, M. K. Ng, and S. Zhang. Block spectral clustering methods for multiple graphs. *Numerical Linear Algebra with Applications*, 24(1):e2075, 2017. doi: 10.1002/nla.2075. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nla.2075.

Chengjun Liu and H. Wechsler. A gabor feature classifier for face recognition. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 270–275 vol.2, July 2001. doi: 10.1109/ICCV.2001.937635. URL https://ieeexplore.ieee.org/abstract/document/937635.

F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi. Tensor decomposition of eeg signals: A brief review. *Journal of Neuroscience Methods*, 248:59 – 69, 2015. ISSN 0165-0270. doi: 10.1016/j.jneumeth.2015.03.018. URL http://www.sciencedirect.com/science/article/pii/S0165027015001016.

K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 321–328. MIT Press, 2007. doi: 10.5555/1390681.1442790. URL http://papers.nips.cc/paper/2972-learning-from-multiple-sources.pdf.

J.-J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. *Statistics and computing*, 18(2):173–183, 2008. doi: 10.1007/s11222-007-9046-7. URL https://link.springer.com/article/10.1007/s11222-007-9046-7.

L. De Lathauwer. A survey of tensor methods. In *2009 IEEE International Symposium on Circuits and Systems*, pages 2773–2776, May 2009. doi: 10.1109/ISCAS.2009.5118377. URL https://ieeexplore.ieee.org/abstract/document/5118377.

L. De Lathauwer, B. De Moor, and J. Vandewall. A multilinear singular value decomposition. *SIAM Journal Matrix Ana. Appl*, 11:1253–1278, 2000. doi: 10.1137/S0895479896305696. URL https://epubs.siam.org/doi/10.1137/S0895479896305696.

W. S. De Sarbo and W. L. Corn. A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification*, pages 249–282, 1988. doi: 10.1007/BF01897167. URL https://link.springer.com/article/10.1007/BF01897167.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977. doi: 10.1111/j.2517-6161.1977.tb01600.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x.

M. Deodhar and J. Ghosh. Scoal: A framework for simultaneous co-clustering and learning from complex data. *ACM Trans. Knowl. Discov. Data*, 4(3), 2010. ISSN 1556-4681. doi: 10.1145/1839490.1839492. URL https://doi.org/10.1145/1839490.1839492.

I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 269–274. Association for Computing Machinery, 2001. ISBN 158113391X. doi: 10.1145/502512.502550. URL https://doi.org/10.1145/502512.502550.

I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 89–98. Association for Computing Machinery, 2003. ISBN 1581137370. doi: 10.1145/956750.956764. URL https://doi.org/10.1145/956750.956764.

T. G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-45014-6. doi: 10.1007/3-540-45014-9_1. URL https://link.springer.com/chapter/10.1007/3-540-45014-9_1.

C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 126–135, 2006. ISBN 1595933395. doi: 10.1145/1150402.1150420. URL https://doi.org/10.1145/1150402.1150420.

C. H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 107–114, 2001. doi: 10.1109/ICDM.2001.989507. URL https://ieeexplore.ieee.org/document/989507.

L. Drumond, S. Rendle, and L. Schmidt-Thieme. Predicting rdf triples in incomplete knowledge bases with tensor factorization. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, page 326–331. Association for Computing Machinery, 2012. ISBN 9781450308571. doi: 10.1145/2245276.2245341. URL https://doi.org/10.1145/2245276.2245341.

B. Du, M. Zhang, L. Zhang, R. Hu, and D. Tao. Pltd: Patch-based low-rank tensor decomposition for hyperspectral images. *IEEE Transactions on Multimedia*, 19(1):67–79, 2017. ISSN 1941-0077. doi: 10.1109/TMM.2016.2608780. URL https://ieeexplore.ieee.org/abstract/document/7565539.

T. J. Durham, M. W. Libbrecht, J. J. Howbert, J. Bilmes, and W. S. Noble. Predictd parallel epigenomics data imputation with cloud-based tensor decomposition. *Nature communications*, 9(1):1402, 2018. doi: 10.1038/s41467-018-03635-9. URL https://www.nature.com/articles/s41467-018-03635-9.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page 226–231. AAAI Press, 1996. doi: 10.5555/3001460.3001507. URL https://dl.acm.org/doi/10.5555/3001460.3001507.

B. S. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis, 5th Edition*. Wiley Series in Probability and Statistics, 2011.

S. Feizi, H. Javadi, and D. Tse. Tensor biclustering. In *Advances in Neural Information Processing Systems 30*, pages 1311–1320. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6730-tensor-biclustering.pdf.

F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering*, 19(3):355–369, 2007. doi: 10.1109/TKDE.2007.46. URL https://ieeexplore.ieee.org/document/4072747.

C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588, 1998. ISSN 1460-2067. doi: 10.1093/comjnl/41.8.578. URL https://ieeexplore.ieee.org/abstract/document/8129683.

T. Frankel. *The Geometry of Physics: An Introduction*. Cambridge University Press, 2012.

T. Franz, A. Schultz, S. Sizov, and S. Staab. Triplerank: Ranking semantic web data by tensor decomposition. In *The Semantic Web - ISWC 2009*, pages 213–228, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. doi: 10.1007/978-3-642-04930-9_14. URL https://link.springer.com/chapter/10.1007/978-3-642-04930-9_14.

J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. A graph-based consensus maximization approach for combining multiple supervised and unsupervised models. volume 25, pages 15–28. 2013. doi: 10.1109/TKDE.2011.206. URL https://ieeexplore.ieee.org/document/6035707.

J. Ghosh. Simultaneous (co)-clustering and modeling for large scale data mining. In *Fall Creek Falls Conference,[online]*, 2009. URL https://computing.ornl.gov/workshops/FallCreek09/presentations/j_ghosh.pdf.

N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the National Conference on Artificial Intelligence*, Proceedings of the National Conference on Artificial Intelligence, pages 439–446. AAAI, 1999. ISBN 0262511061. doi: 10.5555/315149.315352. URL https://dl.acm.org/doi/10.5555/315149.315352.

G. Govaert. La classification croisée. *La revue de Modulad*, pages 9–36, 1983.

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

G. Govaert and M. Nadif. Comparison of the mixture and the classification maximum likelihood in cluster analysis with binary data. *Computational Statistics & Data Analysis*, 23(1): 65 – 81, 1996. ISSN 0167-9473. doi: 10.1016/S0167-9473(96)00021-7. URL http://www.sciencedirect.com/science/article/pii/S0167947396000217.

G. Govaert and M. Nadif. Clustering with block mixture models. *Pattern Recognition*, 36(2):463 – 473, 2003. ISSN 0031-3203. doi: 10.1016/S0031-3203(02)00074-2. URL http://www.sciencedirect.com/science/article/pii/S0031320302000742.

G. Govaert and M. Nadif. An em algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):643–647, 2005. ISSN 1939-3539. doi: 10.1109/TPAMI.2005.69. URL https://ieeexplore.ieee.org/document/1401917.

G. Govaert and M. Nadif. Fuzzy clustering to estimate the parameters of block mixture models. *Soft Computing*, 10(5):415–422, 2006. doi: 10.1007/s00500-005-0502-z. URL https://link.springer.com/article/10.1007/s00500-005-0502-z.

G. Govaert and M. Nadif. Block clustering with bernoulli mixture models: Comparison of different approaches. *Computational Statistics and Data Analysis*, 52(6):3233–3245, 2008. doi: 10.1016/j.csda.2007.09.007. URL https://www.sciencedirect.com/science/article/pii/S0167947307003441.

G. Govaert and M. Nadif. Latent block model for contingency table. *ASMDA 2007 International Conference on Applied Stochastic Models and Data Analysis*, pages 416–425, 2010. doi: 10.1080/03610920903140197. URL https://www.tandfonline.com/doi/abs/10.1080/03610920903140197.

G. Govaert and M. Nadif. *Co-clustering: models, algorithms and applications*. John Wiley & Sons, 2013. URL https://www.wiley.com/en-us/Co+Clustering%3A+Models%2C+Algorithms+and+Applications-p-9781848214736.

G. Govaert and M. Nadif. Mutual information, phi-squared and model-based co-clustering for contingency tables. *Advances in Data Analysis and Classification*, 12(3):455–488, Sep 2018. doi: 10.1007/s11634-016-0274-6. URL https://link.springer.com/article/10.1007/s11634-016-0274-6.

L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013. doi: 10.1002/gamm.201310004. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.201310004.

R. Guigoures, M. Boulle, and F. Rossi. A triclustering approach for time evolving graphs. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops*, ICDMW '12, pages 115–122, 2012. ISBN 978-0-7695-4925-5.

X. Guo, X. Huang, L. Zhang, and L. Zhang. Hyperspectral image noise reduction based on rank-1 tensor decomposition. *ISPRS Journal of Photogrammetry and Remote Sensing*, 83:50 – 63, 2013. doi: 10.1016/j.isprsjprs.2013.06.001. URL https://www.sciencedirect.com/science/article/abs/pii/S092427161300138X.

L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*,

11(9):1074–1085, 1992. ISSN 1937-4151. doi: 10.1109/43.159993. URL https://ieeexplore.ieee.org/document/159993.

B. Hanczar and M. Nadif. Ensemble methods for biclustering tasks. *Pattern Recognition*, 45(11):3938–3949, 2012. doi: 10.1016/j.patcog.2012.04.010. URL https://www.sciencedirect.com/science/article/abs/pii/S0031320312001677.

K. H. Hansen and J. Krarup. Improvements of the held—karp algorithm for the symmetric traveling-salesman problem. *Mathematical Programming*, 7(1):87–96, 1974. doi: 10.1007/BF01585505. URL https://link.springer.com/article/10.1007/BF01585505.

L. Hao, S. Liang, J. Ye, and Z. Xu. Tensord: A tensor decomposition library in tensorflow. *Neurocomputing*, 318:196 – 200, 2018. doi: 10.1016/j.neucom.2018.08.055. URL https://www.sciencedirect.com/science/article/abs/pii/S0925231218310178.

R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973. ISSN 2168-2909. doi: 10.1109/TSMC.1973.4309314. URL https://ieeexplore.ieee.org/document/4309314.

R. Harshman. Foundations of the parafac procedure: Models and conditions for an explanatory factor analysis. *Technical Report UCLA Working Papers in Phonetics 16, University of California, Los Angeles, Los Angeles, CA*, 1970. URL https://pdfs.semanticscholar.org/c5b2/8cae82b14417f1250e58bb241367248e827d.pdf?_ga=2.63100335.169746605.1584553731-1948055760.1584553731.

R. A. Harshman. Parafac2: Mathematical and technical notes. *UCLA working papers in phonetics*, 22(3044):122215, 1972.

R. A. Harshman. Models for analysis of asymmetrical relationships among n objects or stimuli. *First joint meeting of the psychometric society and society of mathematical psychology, McMaster University*, 1994. URL https://www.semanticscholar.org/paper/Models-for-analysis-of-asymmetrical-relationships-n-Harshman/7fbca54aad791e70380e678e668350ce99d80466.

R. A. Harshman and M. E. Lundy. Parafac : parallel factor analysis. *Computational statistics and data analysis*, 18:39–72, 1994. doi: 10.1016/0167-9473(94)90132-5. URL https://www.sciencedirect.com/science/article/pii/0167947394901325.

R. A. Harshman and M. E. Lundy. Uniqueness proof for a family of models sharing features of tucker's three-mode factor analysis and parafac/candecomp. *Psychometrika*, 61(1):133–154, 1996. doi: 10.1007/BF02296963. URL https://link.springer.com/article/10.1007/BF02296963.

J. A. Hartigan. Direct clustering of a data matrix. *Journal of the american statistical association*, 67(337):123–129, 1972.

M. Hašan, E. Velázquez-Armendariz, F. Pellacini, and K. Bala. Tensor clustering for rendering many-light animations. In *Proceedings of the Nineteenth Eurographics Conference on Rendering*, EGSR '08, pages 1105–1114, 2008. doi: 10.1111/j.1467-8659.2008.01248.x. URL https://dl.acm.org/doi/10.1111/j.1467-8659.2008.01248.x.

X. He, D. Cai, H. Liu, and J. Han. Image clustering with tensor representation. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 132–140, 2005. ISBN 1-59593-044-2. doi: 10.1145/1101149.1101169. URL https://dl.acm.org/doi/10.1145/1101149.1101169.

R. Henriques and S. C. Madeira. Triclustering algorithms for three-dimensional data analysis: A comprehensive survey. *ACM Comput. Surv.*, 51(5):95:1–95:43, Sept. 2018. ISSN 0360-0300. doi: 10.1145/3195833. URL https://dl.acm.org/doi/abs/10.1145/3195833.

F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927. doi: 10.1002/sapm192761164. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sapm192761164.

F. L. Hitchcock. Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics*, 7(1-4):39–79, 1928. doi: 10.1002/sapm19287139. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sapm19287139.

V. Hore, A. Viñuela, A. Buil, J. Knight, M. I. McCarthy, K. Small, and J. Marchini. Tensor decomposition for multiple-tissue gene expression experiments. *Nature Genetics*, 48 (9):1546–1718, 2016. doi: 10.1038/ng.3624. URL https://www.nature.com/articles/ng.3624.

E. Hosseini-Asl and J. M. Zurada. Nonnegative matrix factorization for document clustering: A survey. In *Artificial Intelligence and Soft Computing*, 2014. doi: 10.1007/978-3-319-07176-3_63. URL https://link.springer.com/chapter/10.1007/978-3-319-07176-3_63.

S. Janson. Poisson convergence and poisson processes with applications to random graphs. *Stochastic Processes and their Applications*, 26:1 – 30, 1987. doi: 10.1016/0304-4149(87)90048-2. URL https://www.sciencedirect.com/science/article/pii/0304414987900482.

Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. ISSN 1939-3539. doi: 10.1109/34.868688. URL https://ieeexplore.ieee.org/document/868688.

A. Kapteyn, H. Neudecker, and T. Wansbeek. An approach ton-mode components analysis. *Psychometrika*, 51(2):269–275, 1986. doi: 10.1007/BF02293984. URL https://link.springer.com/article/10.1007/BF02293984.

A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys*, 2010. doi: 10.1145/1864708.1864727. URL https://dl.acm.org/doi/10.1145/1864708.1864727.

B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011. doi: 10.1103/PhysRevE.83.016107. URL https://journals.aps.org/pre/abstract/10.1103/PhysRevE.83.016107.

L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Statistics, 1990. URL https://www.wiley.com/en-us/Finding+Groups+in+Data%3A+An+Introduction+to+Cluster+Analysis-p-9780471735786.

H. A. L. Kiers. Towards a standardized notation and terminology in multiway analysis. *JOURNAL OF CHEMOMETRICS*, 14:105–122, 2000.

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *Journal of mathematical psychology*, 51(3):455–500, 2009. doi: 10.1137/07070111X. URL https://epubs.siam.org/doi/abs/10.1137/07070111X?journalCode=siread.

J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic. Tensorly: Tensor learning in python. *J. Mach. Learn. Res.*, 20(1):925–930, Jan. 2019. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=3322706.3322732.

D. Kressner and L. Perisa. Recompression of hadamard products of tensors in tucker format. *SIAM J. Scientific Computing*, 39, 2017. doi: 10.1137/16M1093896. URL https://epubs.siam.org/doi/abs/10.1137/16M1093896?journalCode=sjoce3.

H. Kriegel, P. Kröger, J. Sander, and A. Zimek. *Density-based clustering*. Wiley, 2011a. doi: 10.1007/978-0-387-30164-8_211. URL https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8_211.

H.-P. Kriegel, P. Kröger, I. Ntoutsi, and A. Zimek. Density based subspace clustering over dynamic data. In J. Bayard Cushing, J. French, and S. Bowers, editors, *Scientific and Statistical Database Management*. Springer Berlin Heidelberg, 2011b. doi: 10.1007/978-3-642-22351-8_24. URL https://link.springer.com/chapter/10.1007/978-3-642-22351-8_24.

P. M. Kroonenberg. *Three-mode principal component analysis: Theory and applications*, volume 2. DSWO press, 1983.

R. M. Kumar and K. Sreekumar. A survey on image feature descriptors. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 5(1):7668–7673, 2014. URL http://ijcsit.com/docs/Volume%205/vol5issue06/ijcsit20140506168.pdf.

L. Labiod and M. Nadif. Co-clustering under nonnegative matrix tri-factorization. In *International Conference on Neural Information Processing*, pages 709–717. Springer, 2011. doi: 10.1007/978-3-642-24958-7_82. URL https://link.springer.com/chapter/10.1007/978-3-642-24958-7_82.

Last.fm. Music recommendation service. http://www.last.fm, 2009.

P. Legendre and L. Legendre. *Numerical Ecology, Volume 24, 2nd Edition*. Developments in Environmental Modelling, 1998.

S. Li, W. Wang, H. Qi, B. Ayhan, C. Kwan, and S. Vance. Low-rank tensor decomposition based anomaly detection for hyperspectral imagery. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4525–4529. IEEE, 2015. doi: 10.1109/ICIP.2015.7351663. URL https://ieeexplore.ieee.org/document/7351663.

T. Li and C.-c. Ding. Nonnegative matrix factorizations for clustering: A survey. In *Data Clustering*, pages 149–176. Chapman and Hall/CRC, 2018. doi: 10.1007/978-3-319-07176-3_63. URL https://link.springer.com/chapter/10.1007/978-3-319-07176-3_63.

L.-H. Lim and P. Comon. Multiarray signal processing: Tensor decomposition meets compressed sensing. *Comptes Rendus Mécanique*, 338(6):311 – 320, 2010. ISSN 1631-0721. doi: https://doi.org/10.1016/j.crme.2010.06.005. URL http://www.sciencedirect.com/science/article/pii/S163107211000094X.

G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):171–184, 2013a. ISSN 1939-3539. doi: 10.1109/TPAMI.2012.88. URL https://ieeexplore.ieee.org/abstract/document/6180173.

G. Liu, Z. Lin, S. Yan, J. Sun, and Y. Yu, Yong ; Ma. Properties of the hubert-arabie adjusted rand index. *IEEE Transactions on Pattern Analysis and Machine Intelligences*, 35(1):171 – 184, 2013b. doi: 10.1037/1082-989X.9.3.386. URL https://www.ncbi.nlm.nih.gov/pubmed/15355155.

Y. Liu, T. Yang, and L. Fu. A partitioning based algorithm to fuzzy tricluster. *Mathematical Problems in Engineering*, 2015, 2015.

B. Long, Z. M. Zhang, and P. S. Yu. Co-clustering by block value decomposition. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 635–640, 2005. ISBN 1-59593-135-X.

R. Maclin and D. Opitz. An empirical evaluation of bagging and boosting. *AAAI/IAAI*, 1997: 546–551, 1997. doi: 10.5555/1867406.1867491. URL https://dl.acm.org/doi/10.5555/1867406.1867491.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. 1967. URL https://projecteuclid.org/euclid.bsmsp/1200512992.

A. G. Mahyari, D. M. Zoltowski, E. M. Bernat, and S. Aviyente. A tensor decomposition-based approach for detecting dynamic network states from eeg. *IEEE Transactions on Biomedical Engineering*, 64(1):225–237, 2017. ISSN 1558-2531. doi: 10.1109/TBME.2016.2553960. URL https://ieeexplore.ieee.org/document/7452353.

S. Mankad and G. Michailidis. Biclustering three-dimensional data arrays with plaid models. *Journal of Computational and Graphical Statistics*, 23(4):943–965, 2014. doi: 10.1080/10618600.2013.851608. URL https://www.tandfonline.com/doi/abs/10.1080/10618600.2013.851608?journalCode=ucgs20.

G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, 2000.

K. McLeod, M. Sermesant, P. Beerbaum, and X. Pennec. Spatio-temporal tensor decomposition of a polyaffine motion model for a better analysis of pathological left ventricular dynamics. *IEEE Transactions on Medical Imaging*, 34(7):1562–1575, 2015. doi: 10.1109/TMI.2015.2405579. URL https://ieeexplore.ieee.org/document/7045543.

B. Moberts, A. Vilanova, and J. J. van Wijk. Evaluation of fiber clustering methods for diffusion tensor imaging. In *VIS 05. IEEE Visualization, 2005.*, pages 65–72, 2005. doi: 10.1109/VISUAL.2005.1532779. URL https://ieeexplore.ieee.org/document/1532779.

J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. URL https://www.jstor.org/stable/2098689?seq=1.

F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview, ii. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(6):e1219, 2017. doi: 10.1002/widm.53. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.53.

M. Nadif and G. Govaert. Block clustering of contingency table and mixture model. In *International Symposium on Intelligent Data Analysis*, pages 249–259. Springer, 2005. doi: 10.1007/11552253_23. URL https://link.springer.com/chapter/10.1007/11552253_23.

M. Nadif and G. Govaert. Model-based co-clustering for continuous data. *Ninth International Conference on Machine Learning and Applications (ICMLA)*, 2010. doi: 10.1109/ICMLA.2010.33. URL https://ieeexplore.ieee.org/document/5708830.

R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models*, pages 355–368, 1998. doi: 10.1007/978-94-011-5014-9_12. URL https://link.springer.com/chapter/10.1007/978-94-011-5014-9_12.

O. Nenadic and M. Greenacre. Correspondence analysis in r, with two-and three-dimensional graphics: The ca package. *Journal of statistical software*, 20(3), 2007. doi: 10.18637/jss.v020.i03. URL https://www.jstatsoft.org/article/view/v020i03.

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004. doi: 10.1103/PhysRevE.69.026113. URL https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

M. Nickel, T. V., and K. H. P. A three-way model for collective learning on mluti-relational data. *International Conference on Machine Learning (ICML), Bellevue, WA, USA*, 2011. doi: 10.5555/3104482.3104584. URL https://dl.acm.org/doi/10.5555/3104482.3104584.

F. Nie, J. Li, X. Li, et al. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, pages 2564–2570, 2017. doi: 10.5555/3172077.3172245. URL https://dl.acm.org/doi/abs/10.5555/3172077.3172245.

K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association*, 96(455):1077–1087, 2001. doi: 10.1198/016214501753208735. URL https://www.tandfonline.com/doi/abs/10.1198/016214501753208735.

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

K. Pearson. On the probability that two independent distributions of frequency are really samples from the same population. *Biometrika*, 8(1/2):250–254, 1911. doi: 10.2307/2331453. URL https://www.jstor.org/stable/pdf/2331453.pdf?seq=1.

W. Penga and T. Lib. Tensor clustering via adaptive subspace iteration. *Intelligent Data Analysis*, 15:695–713, 2011. doi: 10.3233/IDA-2011-0490. URL https://content.iospress.com/articles/intelligent-data-analysis/ida00490.

M. Qiao, J. Yu, W. Bian, Q. Li, and D. Tao. Improving stochastic block models by incorporating power-law degree characteristic. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2620–2626, 2017. doi: 10.24963/ijcai.2017/365. URL https://doi.org/10.24963/ijcai.2017/365.

D. Rafailidis and P. Daras. The tfc model: Tensor factorization and tag clustering for item recommendation in social tagging systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(3):673–688, 2013. doi: 10.1109/TSMCA.2012.2208186. URL https://ieeexplore.ieee.org/document/6301770.

F. Ricci, L. Rokach, and B. Shapira. *Introduction to Recommender Systems Handbook*. Springer US, 2011. URL http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf.

G. Ricci, M. De Gemmis, and G. Semeraro. Matrix and tensor factorization techniques applied to recommender systems: a survey. *International Journal of Computer and Information Technology*, 1(1):(2277–0764), 2012. URL https://pdfs.semanticscholar.org/1b97/7c27db5fcf669f8cd63dfbdad7e6b85fa179.pdf.

F. Role, S. Morbieu, and M. Nadif. Coclust: A python package for co-clustering. *Journal of Statistical Software*, 88(7):1–29, 2019. doi: 10.18637/jss.v088.i07. URL https://www.jstatsoft.org/article/view/v088i07.

S. Saha, C. A. Murthy, and S. K. Pal. Classification of web services using tensor space model and rough ensemble classifier. In A. An, S. Matwin, Z. W. Raś, and D. Ślęzak, editors, *Foundations of Intelligent Systems*, pages 508–513. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-68123-6_55. URL https://link.springer.com/chapter/10.1007/978-3-540-68123-6_55.

A. Salah and M. Nadif. Social regularized von mises–fisher mixture model for item recommendation. *Data Mining and Knowledge Discovery*, 31(5):1218–1241, 2017. doi: 10.1007/s10618-017-0499-9. URL https://link.springer.com/article/10.1007/s10618-017-0499-9.

A. Salah and M. Nadif. Directional co-clustering. *Advances in Data Analysis and Classification*, pages 1–30, 2018.

A. Salah, M. Ailem, and M. Nadif. Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

R. E. Schapire. The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification*, pages 149–171. Springer, 2003. doi: 10.1007/978-0-387-21579-2_9. URL https://link.springer.com/chapter/10.1007/978-0-387-21579-2_9.

J. Schepers, I. van Mechelen, and E. Ceulemans. Three-mode partitioning. *Computational Statistics & Data Analysis*, 51(3):1623 – 1642, 2006. doi: 10.1016/j.csda.2006.06.002. URL https://www.sciencedirect.com/science/article/pii/S0167947306001927.

G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

A. J. Scott and M. J. Symons. Clustering methods based on likelihood ratio criteria. *Biometrics*, 27(2):387–397, 1971. doi: 10.2307/2529003. URL https://www.jstor.org/stable/2529003?seq=1.

H. Shan and A. Banerjee. Bayesian co-clustering. In *2008 Eighth IEEE International Conference on Data Mining*, pages 530–539. IEEE, 2008. doi: 10.1109/ICDM.2008.91. URL https://ieeexplore.ieee.org/document/4781148.

R. Sibson. SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973. doi: 10.1093/comjnl/16.1.30. URL https://academic.oup.com/comjnl/article/16/1/30/434805.

N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

R. Sokal, C. Michener, and U. of Kansas. *A Statistical Method for Evaluating Systematic Relationships*. University of Kansas science bulletin. University of Kansas, 1958.

H. Späth. Clusterwise linear regression. *Computing*, pages 367–373, 1979. doi: 10.1007/BF02265317. URL https://link.springer.com/article/10.1007/BF02265317.

S. Sra, S. Jegelka, and A. Banerjee. Approximation algorithms for bregman clustering co-clustering and tensor clustering, 2008. URL http://hdl.handle.net/11858/00-001M-0000-0013-C75F-8.

A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002. doi: 10.1162/153244303321897735. URL https://dl.acm.org/doi/10.1162/153244303321897735.

P. Symeonidis. Matrix and tensor decomposition in recommender systems. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 429–430, 2016. ISBN 978-1-4503-4035-9. doi: 10.1145/2959100.2959195. URL https://dl.acm.org/doi/abs/10.1145/2959100.2959195.

M. J. Symons. Clustering criteria and multivariate normal mixtures. *Biometrics*, 37(1):35–43, 1981. doi: 10.2307/2530520. URL https://www.jstor.org/stable/2530520?seq=1.

J. Tang, X. Shu, G. Qi, Z. Li, M. Wang, S. Yan, and R. Jain. Tri-clustered tensor completion for social-aware image tag refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1662–1674, 2017. doi: 10.1109/TPAMI.2016.2608882. URL https://ieeexplore.ieee.org/document/7565563.

J. Tang, X. Shu, Z. Li, Y.-G. Jiang, and Q. Tian. Social anchor-unit graph regularized tensor completion for large-scale image retagging. *IEEE transactions on pattern analysis and machine intelligence*, 2019. doi: 10.1109/TPAMI.2019.2906603. URL https://ieeexplore.ieee.org/document/8673651.

W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In *2009 Ninth IEEE International Conference on Data Mining*, pages 1016–1021. IEEE, 2009. doi: 10.1109/ICDM.2009.125. URL https://ieeexplore.ieee.org/abstract/document/5360349?section=abstract.

A. B. Tchagang, S. Phan, F. Famili, H. Shearer, P. Fobert, Y. Huang, J. Zou, D. Huang, A. Cutler, Z. Liu, and Y. Pan. Mining biological information from 3d short time-series

gene expression data: the optricluster algorithm. *BMC Bioinformatics*, 13(54), 2012. doi: 10.1186/1471-2105-13-54. URL https://www.ncbi.nlm.nih.gov/pubmed/22475802.

G. Thomas and S. Merugu. A scalable collaborative filtering framework based on co-clustering. *Fifth IEEE International Conference on Data Mining*, 2005. doi: 10.1109/ICDM.2005.14. URL https://ieeexplore.ieee.org/document/1565742.

L. R. Tucker. Implications of factor analysis of three-way matrices for measurement of change. *Problems in measuring change, C. W. Harris, ed., University of Wisconsin Press*, pages 122–137, 1963.

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

T. Turchet. Tgap dÉchets : Le gouvernement propose une rÉforme ambitieuse dans le projet de loi de finances pour 2019. *Zerowaste France*, 2018. URL https://www.zerowastefrance.org/tgap-dechets-reforme-plf-2019/.

M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision*, pages 447–460. Springer, 2002. doi: 10.5555/645315.649173. URL https://dl.acm.org/doi/10.5555/645315.649173.

S. Vega-Pons and J. Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372, 2011. doi: 10.1142/S0218001411008683. URL https://www.worldscientific.com/doi/abs/10.1142/S0218001411008683.

M. A. Veganzones, J. E. Cohen, R. Cabral Farias, J. Chanussot, and P. Comon. Nonnegative tensor cp decomposition of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 54(5):2577–2588, 2016. doi: 10.1109/TGRS.2015.2503737. URL https://ieeexplore.ieee.org/document/7360181.

A. Veit, M. Nickel, S. Belongie, and L. Maaten. Separating self-expression and visual content in hashtag supervision. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL https://arxiv.org/abs/1711.09825.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec 2007. URL https://arxiv.org/abs/0711.0189.

D. Vu and M. Aitkin. Variational algorithms for biclustering models. *Computational Statistics and Data Analysis*, pages 12–24, 2015. doi: 10.1016/j.csda.2015.02.015. URL https://www.sciencedirect.com/science/article/pii/S0167947315000560.

H. Wang, F. Nie, H. Huang, and C. Ding. Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation. In *2011 IEEE 11th international conference on data mining*, pages 774–783. IEEE, 2011.

X. Wang, C. Yang, and S. Mao. Tensorbeat: Tensor decomposition for monitoring multiperson breathing beats with commodity wifi. *ACM Trans. Intell. Syst. Technol.*, 9 (1):8:1–8:27, Sept. 2017. ISSN 2157-6904. doi: 10.1145/3078855. URL https://dl.acm.org/doi/10.1145/3078855.

Y. Wang, J. Peng, Q. Zhao, Y. Leung, X. Zhao, and D. Meng. Hyperspectral image restoration via total variation regularized low-rank tensor decomposition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(4):1227–1243, 2018. doi: 10.1109/JSTARS.2017.2779539. URL https://ieeexplore.ieee.org/abstract/document/8233403.

J. J. H. Ward. Hierarchical grouping to optimize an objective function. 1963. doi: 10.1080/01621459.1963.10500845. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845.

H. Wermser, A. Rettinger, and V. Tresp. Modeling and learning context-aware recommendation scenarios using tensor decomposition. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 137–144, 2011. doi: 10.1109/ASONAM.2011.56. URL https://ieeexplore.ieee.org/document/5992573.

T. Wu, A. R. Benson, and D. F. Gleich. General tensor spectral co-clustering for higher-order data. In *Advances in Neural Information Processing Systems 29*, pages 2559–2567. Curran Associates, Inc., 2016. doi: 10.5555/3157382.3157385. URL https://dl.acm.org/doi/10.5555/3157382.3157385.

X. Wu, R. Zurita-Milla, E. Izquierdo Verdiguier, and M.-J. Kraak. Triclustering georeferenced time series for analyzing patterns of intra-annual variability in temperature. *Annals of the American Association of Geographers*, 108(1):71–87, 2018. doi: 10.1080/24694452.2017.1325725. URL https://www.tandfonline.com/doi/citedby/10.1080/24694452.2017.1325725?scroll=top&needAccess=true.

H. Yan, K. Paynabar, and J. Shi. Image-based process monitoring using low-rank tensor decomposition. *IEEE Transactions on Automation Science and Engineering*, 12(1):216–227, 2014. doi: 10.1109/TASE.2014.2327029. URL https://ieeexplore.ieee.org/document/6855374.

J. Yoo and S. Choi. Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds. *Information Processing & Management*, 46(5):559 – 570, 2010. doi: 10.1016/j.ipm.2009.12.007. URL https://www.sciencedirect.com/science/article/abs/pii/S0306457310000038.

X. Yu, G. Yu, J. Wang, and C. Domeniconi. Co-clustering ensembles based on multiple relevance measures. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2019. ISSN 2326-3865. doi: 10.1109/TKDE.2019.2942029. URL https://ieeexplore.ieee.org/document/8840892.

C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao. Low-rank tensor constrained multiview subspace clustering. In *Proceedings of the IEEE international conference on computer vision*,

pages 1582–1590, 2015. doi: 10.1109/ICCV.2015.185. URL https://ieeexplore.ieee.org/document/7410542.

J. Zhang, Y. Han, and J. Jiang. Tucker decomposition-based tensor learning for human action recognition. *Multimedia Systems*, 22(3):343–353, Jun 2016. doi: 10.1007/s00530-015-0464-7. URL https://link.springer.com/article/10.1007/s00530-015-0464-7.

W. Zhang, J. Han, and S. Deng. Heart sound classification based on scaled spectrogram and tensor decomposition. *Expert Systems with Applications*, 84:220 – 231, 2017. doi: 10.1016/j.eswa.2017.05.014. URL https://www.sciencedirect.com/science/article/abs/pii/S0957417417303305.

X. Zhang, G. Wen, and W. Dai. A tensor decomposition-based anomaly detection algorithm for hyperspectral image. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):5801–5820, 2016. ISSN 1558-0644. doi: 10.1109/TGRS.2016.2572400. URL https://ieeexplore.ieee.org/document/7493677.

Z.-K. Zhang, T. Zhou, and Y.-C. Zhang. Tag-aware recommender systems: A state-of-the-art survey. *JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY*, 26(5):(767–777), 2011. doi: 10.1007/s11390-011-0176-1. URL https://link.springer.com/article/10.1007/s11390-011-0176-1.

Z.-Y. Zhang, T. Li, and C. Ding. Non-negative tri-factor tensor decomposition with applications. *Knowledge and Information Systems*, 34(2):243–265, Feb 2013. doi: 10.1007/s10115-011-0460-y. URL https://link.springer.com/article/10.1007/s10115-011-0460-y.

L. Zhao and M. J. Zaki. tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. In *In Proc. of the 2005 ACM SIGMOD international conference on Management of data*, pages 694–705. ACM Press, 2005. doi: 10.1145/1066157.1066236. URL https://dl.acm.org/doi/10.1145/1066157.1066236.