THESE EN COTUTELLE

Pour l'obtention du grade de Docteur délivré par

L'UNIVERSITE DE POITIERS

Ecole Doctorale Sciences et Ingénierie des Systèmes, Mathématiques, Informatique

et

L'Université Libanaise

Ecole Doctorale des Sciences et Technologies

Spécialité: Mathématiques Appliquées

présentée par

Abir El Haj

STOCHASTIC BLOCKMODELS, CLASSIFICATIONS AND APPLICATIONS

Directeurs de Thèse : **Yousri Slaoui et Zaher Khraibani**
Codirecteur de Thèse : **Pierre-Yves Louis**

Présentée et soutenue publiquement le
29 novembre 2019

COMPOSITION DU JURY

Rapporteurs :	**Sophie Donnet**	Chargée de Recherches INRA (HDR), AgroParisTech
	Stéphane Robin	Directeur de Recherches, INRA, AgroParisTech
Examinateurs :	**Ghislaine Gayraud**	Professeur, Université de technologie de Compiègne
	Mohammad Kacim	Maître de conférences, Université Saint Esprit de Kaslik
	Zaher Khraibani	Professeur, Université Libanaise
	Pierre-Yves Louis	Maître de conférences (HDR), Université de Poitiers
	Joseph Ngatchou-Wandji	Professeur, Université de Lorraine
	Yousri Slaoui	Maître de conférences (HDR), Université de Poitiers

DÉDICACE

Aux plus chers parents qui ont toujours sacrifié pour mon succès et mon bonheur, et qui ont tout offert pour mon plaisir et mon progrès dont ils cueillent le fruit de ce sacrifice et ce don inépuisables à travers ce travail qui constitue un pas supplémentaire vers mon succès auquel ils ont longtemps rêvé et souhaité.

A Samir EL HAJ et Rouba FEREKH, Je dédie ce travail avec mon grand merci et mes francs souhaits qu'il leur plait et qu'ils en soient fiers.

Remerciements

J'ai essayé à travers ces quelques lignes d'exprimer ma reconnaissance ainsi que les sentiments d'amour et d'attachement que je vous porte.

Je remercie très fortement :

Mes directeurs de thèse Dr. Yousri Slaoui et Pr. Zaher Khraibani ainsi que mon Co-directeur Dr. Pierre-yves Louis pour la confiance et la liberté qu'ils m'ont accordées, leur support et leur soutien pendant toutes ces années et leur aide pour que ce mémoire aboutisse de la meilleure façon possible.

Dr. Sophie Donnet, Chargée de recherches INRA (HDR), AgroParisTech, Pr. Stéphane Robin, Directeur de recherche, INRA, AgroParisTech, pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse, ainsi que la présidente du jury Pr. Ghislaine Gayraud, Professeur à l'Université de technologie de Compiègne, Dr. Mohammad Kacim, Maître de conférence à l'Université Saint Esprit de Kaslik et Pr. Joseph Ngatchou-Wandji, Professeur à l'Université de Lorraine pour l'honneur qu'il m'ont fait d'être examinateurs dans mon jury de thèse.

La directrice de notre Laboratoire Pr. Alessandra Sarti et les directeurs des écoles doctorales Pr. Samuel Boissière et Pr. Mohamad Khalil pour leurs encouragements et leurs disponibilité et gentillesse.

Dr. Cyril Perret et Dr. Nelly Robin pour toutes nos discussions et leur aide précieuse qui m'ont accompagné tout au long de ma thèse, ce travail n'aurait pu être mené à bien sans vos données précieuses et votre

disponibilité.

Je remercie infiniment mes sœurs Jana et Dima, mon frère Abdel Kader pour les beaux sourires, les jolis souvenirs, les longues soirées et pour le soutien au cours de ces trois années, sans votre aide, je n'en serais pas là aujourd'hui.

Mes petits neveux et nièces : Lyn, Karim, Lea et Jad, vous aurez toujours la plus grande place dans mon cœur, les mots me manquent pour vous décrire mon bonheur lorsque je suis avec vous.

A titre plus personnel, Je remercie chaleureusement mon fiancé, Mohamad, pour l'encouragement et la confiance qu'il m'a témoignée, pour le soutien moral ininterrompu surtout pendant les derniers mois de la thèse.

J'adresse un grand merci à Mostafa Koubar, Aida Alamé, Fatima, Neamat et Abdel Ghani pour les bons moments que nous avons passés ensemble pendant l'été dernier.

Je garde un remerciement spécial à Ahmad, l'ami qui était toujours à coté de moi dans les moments difficiles de la thèse ainsi qu'un remerciement à Youssef et Fayrouz, pour les plus beaux moments que je n'oublierai jamais.

Je tiens à remercier :
Sahar, Fatima, Salima, Chayma, Ziad, Mohamad et Achraf pour les moments inoubliables que nous avons passés ensemble et pour leur présence à coté de moi quand j'avais besoin.

Je remercie également Hiba et Nada, mes amies qui ont laissé des jolies traces et des beaux souvenirs dans ma mémoire, l'amitié n'a pas de prix !

Merci mes amis du labo (Carlos, Angelique, Meghdad, Pietro, Antoine, Simone, Amine et Rachad) pour la merveilleuse ambiance dans le laboratoire.

Enfin, je remercie toutes les personnes qui m'ont aidé directement par leur conseils minutieux ou indirectement en m'encourageant à mieux faire à chaque pas.

Je regrette de ne pouvoir qu'utiliser ce petit mot pour exprimer ce GRAND MERCI.

Résumé

Cette thèse de doctorat porte sur l'analyse de réseaux pondérés, graphes finis où chaque arête est associée à un poids représentant l'intensité de sa force. Nous introduisons une extension du modèle à blocs stochastiques (SBM) binaire, appelée modèle à blocs stochastiques binomial (bSBM). Cette question est motivée par l'étude des réseaux de co-citations dans un contexte de fouille de textes où les données sont représentées par un graphe. Les noeuds sont des mots et chaque arête joignant deux mots est pondérée par le nombre de documents inclus dans le corpus citant simultanément cette paire de mots. Nous développons une méthode d'inférence basée sur l'algorithme espérance maximisation variationnel (VEM) pour estimer les paramètres du modèle proposé ainsi que pour classifier les mots du réseau. Puis nous adoptons une méthode qui repose sur la maximisation d'un critère ICL (en anglais integrated classification likelihood) pour sélectionner le modèle optimal et le nombre de clusters. D'autre part, nous développons une approche variationnelle pour traiter le réseau et nous comparons les deux approches. Des applications à des données réelles sont adoptées pour montrer l'efficacité des deux méthodes ainsi que pour les comparer. Enfin, nous développons un SBM pour traiter les réseaux ayant des vecteurs de poids associés aux noeuds. Nous motivons cette méthode par une application qui vise au développement d'un outil d'aide à la spécification de différents traitements cognitifs réalisés par le cerveau lors de la préparation à l'écriture.

Mots clés. Modèle à blocs stochastiques binomial ; Classification ; Fouille de texte ; Inférence variationnelle ; Réseaux pondérés ; Inférence bayésienne variationnelle ; Modèle à blocs stochastiques avec des noeuds pondérés attribués ; Données EEG.

Abstract

This PhD thesis focuses on the analysis of weighted networks, where each edge is associated to a weight representing its strength. We introduce an extension of the binary stochastic block model (SBM), called binomial stochastic block model (bSBM). This question is motivated by the study of co-citation networks in a context of text mining where data is represented by a graph. Nodes are words and each edge joining two words is weighted by the number of documents included in the corpus simultaneously citing this pair of words. We develop an inference method based on a variational maximization algorithm (VEM) to estimate the parameters of the model as well as to classify the words of the network. Then, we adopt a method based on maximizing an integrated classification likelihood (ICL) criterion to select the optimal model and the number of clusters. Otherwise, we develop a variational approach to analyze the given network. Then we compare the two approaches. Applications based on real data are adopted to show the effectiveness of the two methods as well as to compare them. Finally, we develop a SBM model to deal with node-weighted networks. We motivate this approach by an application that aims at the development of a tool to help the specification of different cognitive treatments performed by the brain during the preparation of the writing.

Keywords. Binomial Stochastic blockmodel ; Clustering ; Text mining ; Variational inference ; Weighted networks ; Variational Bayesian inference ; Stochastic blockmodel with attributed weighted nodes ; EEG data.

Table des matières

Introduction Générale en Français **7**

1 General Introduction **11**

1.1 Introduction 11

1.2 Classical Clustering Algorithms in General Framework 13

1.2.1 k-means clustering algorithm 13

1.2.2 Hierarchical clustering algorithm 14

1.2.3 Spectral clustering algorithm 17

1.3 Community Detection in Networks 17

1.3.1 Community detection with hierarchical clustering algorithms 18

1.3.2 Community detection with `K-means` clustering algorithms . . 19

1.3.3 Community detection with spectral clustering algorithms . . 19

1.4 Networks Characteristics 19

1.4.1 Assortative and disassortative mixing 20

1.4.2 Characteristics of nodes in networks 21

1.4.3 Characteristics of edges in networks 21

1.4.4 Global characteristics in networks 22

1.5 Erdös-Rényi Random Graph Model 22

1.6 Stochastic Blockmodel for Binary Graphs 24

1.6.1 Notations and symbols of the model 24

1.6.2 Generation of the stochastic blockmodel data's 25

1.6.3 Maximum likelihood and log-likelihood 26

1.6.4 Log-likelihood of the complete and incomplete data 26

1.6.5 Log-likelihood of the incomplete data 27

1.6.6 Expectation maximization algorithm . . . 28
1.6.7 Advantages and disadvantages of the EM algorithm . . . 28
1.6.8 Variational expectation maximization inference . . . 29
1.6.9 Variational algorithm . . . 31
1.6.10 Integrated complete data likelihood . . . 33
1.6.11 Initialization criteria of the algorithm . . . 33
1.6.12 Stopping criteria of the algorithm . . . 34

Introduction en Français du Chapitre 2 **35**

2 Estimation in a Binomial Stochastic Blockmodel **39**
2.1 Introduction . . . 39
2.2 Specification and Notations of the Model . . . 41
2.3 Generation of the Stochastic Block Model data's . . . 41
2.4 Inference in the Binomial Stochastic Block Model . . . 42
2.4.1 Likelihood of the complete data . . . 42
2.4.2 Likelihood of the incomplete data . . . 43
2.4.3 Variational inference . . . 44
2.4.4 Algorithm of resolution . . . 47
2.5 Integrated Classification Likelihood (ICL) . . . 48
2.6 Numerical experiments . . . 49
2.6.1 Simulated data . . . 49
2.6.2 Co-citation networks . . . 54
2.6.3 Social network : a benchmark dataset . . . 58
2.7 SBM with Poisson distributed weights . . . 60
2.7.1 Generation of the Poisson SBM data's . . . 61
2.7.2 Likelihood of the complete data . . . 61
2.7.3 Variational EM inference . . . 62
2.7.4 Model selection . . . 64
2.7.5 Numerical comparison . . . 65

Introduction en Français du Chapitre 3 **71**

3 Variational Bayesian Inference in binomial SBM **75**
3.1 Introduction . . . 75
3.1.1 Motivation . . . 76
3.2 Definition of the Model . . . 77
3.3 Variational Bayesian Approach . . . 78
3.4 Estimation in Bayesian SBM . . . 79
3.4.1 Variational Bayesian algorithm . . . 87
3.5 Model Selection . . . 87
3.6 Numerical Experiments . . . 89
3.6.1 Simulated data . . . 89
3.6.2 Co-citation networks . . . 91
3.6.3 Social network : a benchmark dataset . . . 95
3.7 Application : Co-citation networks in statistical text mining . . . 97

Introduction en Français du Chapitre 4 **101**

4 Clustering in Attributed Weighted Nodes Network **105**
4.1 Introduction . . . 105
4.1.1 Motivation . . . 106
4.2 The Model . . . 107
4.3 Generation of Stochastic Blockmodel Data's . . . 108
4.4 Variational Inference . . . 109
4.5 Selection Criterion . . . 115
4.6 Application to EEG Data . . . 115

5 Conclusion et Perspectives **119**

Bibliographie **123**

Introduction Générale en Français

Introduction

Les réseaux sont utilisés pour modéliser les interactions entre un ensemble d'entités. Ils sont devenus parmi les outils les plus puissants pour l'analyse moderne des données. Plusieurs auteurs ont récemment développé des modèles et des algorithmes pour l'analyse et le traitement des réseaux. Parmi ces modèles il y a le modèle à blocs stochastiques (`SBM`) proposé par Anderson et al. [1992] et Holland et al. [1983]. C'est un modèle de graphe aléatoire probabiliste qui vise à produire des classes, appelées blocs, ou plus généralement des amas dans les réseaux. Ce modèle a été utilisé dans plusieurs domaines tels que les réseaux et les sciences de la biologie (Fortunato [2010], Porter et al. [2009]) ainsi que dans les statistiques et l'apprentissage automatique (Goldenberg et al. [2010]). Ce modèle est une généralisation du modèle d'Erdös-Réyni proposé par Erdös ans Rényi [1960] en utilisant une structure latente sur les noeuds. Dans ce modèle, les noeuds du réseau sont regroupés dans des blocs disjoints de manière à ce que les noeuds appartenant au même bloc ont la même probabilité de connexion entre eux. De plus, tous ces noeuds ont la même probabilité de connexion avec un autre noeud appartenant à un autre bloc et la probabilité d'existence d'une arête entre deux noeuds dépend seulement des blocs dans lesquels les deux noeuds se trouvent.

Mariadassou et al. [2010] ont proposé une généralisation du modèle `SBM` pour traiter les graphes aléatoires pondérés. Jernite et al. [2014] ont traité le modèle `SBM` avec des arêtes catégorielle, Airoldi et al. [2008] et Latouche et al. [2011] se sont concentrés sur le modèle `SBM` avec des clusters superposés. Plus récemment, Yang et al. [2011], Xu and Hero [2013], Zreik et al. [2017] et Matias and Miel [2017] ont étendu le modèle pour traiter le cas des réseaux dynamiques dans lesquels ils évoluent au cours du temps et Barbillon et al. [2017] ont traité le cas des réseaux multiplex, où plusieurs arêtes peuvent exister entre une paire de noeuds. Ces arêtes représentent les différents types de relation entre ces noeuds.

Plusieurs auteurs se sont concentrés sur l'estimation des paramètres dans le modèle `SBM`. Tout d'abord, Snijders and Nowicki [1997] ont proposé une inférence de maximum de vraisemblance basée sur l'algorithme espérance maximisation (`EM`)

pour estimer les probabilités de connexion entre les noeuds et pour prédire les blocs dans le modèle SBM ayant seulement deux blocs. Ensuite, Nowicki and Snijders [2001] ont généralisé le travail précédent pour traiter le modèle SBM avec un nombre de bloc arbitraire en utilisant une approche bayésienne fondée sur l'échantillonnage de Gibbs. Puisque l'algorithme EM nécessite le calcul de la distribution des étiquettes Z conditionnellement aux observations X, ce qui est généralement impossible à traiter étant donné que les arêtes du réseau ne sont pas indépendantes, Daudin et al. [2008] et Jaakola [2000] ont introduit des méthodes approximatives basées sur une approche variationnelle pour estimer les paramètres et classifier les noeuds. Ils ont utilisé l'algorithme espérance maximisation variationnel (VEM). De plus, Latouche et al. [2012] ont utilisé une inférence bayésienne variationnelle basée sur l'algorithme EM variationnel Bayes (VBEM), alors que Nowicki and Snijders [2001] ont utilisé l'algorithme d'échantillonnage de Gibbs.

Dans la plupart des méthodes déjà traitées dans ce contexte, nous soulignons que le modèle SBM est limité aux réseaux binaires, dans lesquels les arêtes ne sont pas pondérées. Vu que la plupart des réseaux sont pondérés, Thomas and Blitzstein [2011] ont proposé d'appliquer un seuil aux arêtes pondérées. Cette méthode n'est pas efficace puisqu'elle produit des graphes binaires dans lesquels seulement une partie des informations pertinentes sera conservée et les autres seront détruites. Cependant, Mariadassou et al. [2010], Karrer and Newman [2011] et Ball et al. [2011] ont traité le cas des modèles SBM pondérés sans seuillage. Pour cela, ils ont introduit les modèles SBM avec des arêtes pondérées distribuées selon une loi de Poisson.

Les chapitres 2 et 3 traitent le cas des réseaux de co-citations dans un contexte de fouille de texte. Ces réseaux sont composés de mots qui représentent les noeuds du réseau, et d'arêtes joignant chaque paire de mots . Chaque arête est associée à une valeur entière représentant la capacité ou la force de liaison entre les mots. Ces réseaux sont alors pondérés en fonction du nombre de documents dans le corpus considéré citant simultanément cette paire de mots.

Dans le chapitre 2, nous développons un modèle SBM avec des arêtes pondérées distribuées selon une loi binomiale. Cette distribution binomiale a pour paramètres m et $(\pi_{qr})_{q,r}$. Le paramètre m représente le nombre maximale de documents dans le corpus considéré alors que $(\pi_{qr})_{q,r}$ représente la matrice de probabilité de connexion entre les deux clusters q et r. Puis, nous utilisons l'algorithme espérance maximisation variationnel (VEM) pour estimer les paramètres du modèle ainsi que pour classifier les termes présents dans les documents du corpus. Nous adoptons ensuite un critère ICL (en anglais integrated classification likelihood) pour sélectionner le nombre optimal de clusters. Afin de pouvoir valider l'efficacité de notre approche, nous considérons dans un premier temps des données simulées puis dans un second temps des données réelles. Nous comparons aussi notre approche avec le modèle

SBM avec des arêtes pondérées distribuées selon une loi de Poisson (PSBM).

Dans le chapitre 3, nous considérons le modèle SBM avec des arêtes pondérées distribuées selon la loi binomiale en utilisant cette fois la méthode espérance maximisation variationnelle bayésienne (VBEM). Cette méthode nous permet d'estimer les paramètres du modèle proposé. De plus, nous sélectionnons le modèle correspondant au nombre optimal de clusters en utilisant le critère ILvb (en anglais *integrated likelihood variational Bayes*). Nous reprenons les mêmes données introduites dans le chapitre 2 afin de pouvoir comparer les résultats obtenus en utilisant cette approche avec ceux obtenus en utilisant l'approche VEM. D'autre part, nous développons une application sur des données migratoires en introduisant un corpus d'entretiens avec des mineurs migrants, de la région subsaharienne à la côte européenne méditerranéenne. Ces mineurs migrants ont accepté de répondre à un entretien semi-dirigé[1]. Leurs certificats ont été mis dans des textes numérisés constituant le corpus. Le réseau étudié est constitué de 25 termes (parmi les plus fréquents) liés par des arêtes. A chaque arête joignant un couple de termes, est associé le nombre d'entretiens où les deux termes sont utilisés conjointement. Enfin, nous comparons les résultats obtenus en appliquant le VBEM et le VEM.

Dans le chapitre 4, nous traitons le cas des réseaux binaires avec des vecteurs de poids associés au noeuds. L'objectif de ce chapitre est de spécifier les différents traitements cognitifs réalisés par le cerveau lors de la préparation de l'écriture à partir de l'activité électrique produite par les neurones du cerveau et enregistrée par l'électroencéphalogramme. De plus, il a pour objectif d'explorer l'évolution de l'intensité moyenne des clusters au cours de temps en classifiant les 128 électrodes obtenues par les enregistrements électro-encéphalographique (EEG). Le réseau étudié est constitué de 128 électrodes. Chaque électrode correspond à un noeud. De plus, chaque noeud est associé à un vecteur de poids représentant la différence absolue entre l'intensité du signal de l'électrode et celle des électrodes voisines. Le voisinage est défini par rapport aux positions des électrodes sur le bonnet. Ce sont les électrodes proches spatialement. D'autre part, puisque l'intensité électrique peut être positive ou négative, nous attribuons un signe pour chaque arête joignant une paire d'électrodes. Ce signe est positif si la valence de l'intensité des deux noeuds est la même (+/+ ou -/-) et négatif si la valence est différente pour les deux noeuds (+/- ou -/+). Le réseau étudié est alors un réseau binaire ayant des poids associés aux noeuds. Nous développons un modèle SBM afin de classifier les noeuds du réseau étudié. Ce modèle prend deux matrices comme données d'entrées, l'une est la matrice d'adjacence du graphe binaire et l'autre est la matrice de poids associés aux noeuds. Nous développons ensuite l'algorithme espérance maximisation variationnel pour estimer les paramètres du modèle proposé ainsi

1. Expérience réalisée par by N. Robin, Géographe - chargée de recherches (HDR), CEPED, UMR 196 (Paris Descartes - IRD), hébergée à MIGRINTER (CNRS), UMR.

que de classifier les sommets pondérés.

Dans le chapitre 5, nous développons une conclusion générale de la thèse puis nous présentons les travaux de recherche futurs et les perspectives.

Les chapitres 2, 3 et 4 font l'objet d'un pre-print soumis pour publication.

Structure du Chapitre

Ce chapitre est une introduction générale. En effet, dans la section 1.2, nous introduisons des algorithmes classiques de classification dans un cadre général puis dans la section 1.3, nous définissons la détection des communautés dans les réseaux et nous développons les algorithmes classiques de classification pour les données de réseaux. Dans la section 1.4, nous développons des différentes statistiques des réseaux. Dans la section 1.5, nous développons le modèle d'Erdös-Rényi alors que dans la section 1.6, nous développons le modèle à blocs stochastiques pour les réseaux binaires. En effet, dans la sous section 1.6.1, nous introduisons quelques notations et quelques symboles utilisés alors que dans la sous section 1.6.2, nous définissons le modèle à blocs stochastiques pour les réseaux binaires. Dans les sous sections 1.6.3, 1.6.4 et 1.6.5, nous développons la vraisemblance des données complètes et incomplètes puis dans les sous sections 1.6.6 et 1.6.7, nous introduisons l'algorithme espérance maximisation. Nous réalisons une inférence du modèle en utilisant l'algorithme espérance maximisation variationnel dans la sous section 1.6.8 puis nous introduisons l'algorithme de résolution dans la sous section 1.6.9. Dans la sous section 1.6.10, nous introduisons un critère de sélection du modèle. Enfin, dans les sous sections 1.6.11 et 1.6.12, nous adoptons des critères d'initialisation et d'arrêt de l'algorithme proposé.

Chapitre 1

General Introduction

1.1 Introduction

Networks are used to model interactions between a set of entities. They became one of the most powerful tools for modern data analysis. Several authors have recently developed models and algorithms for network analysis and processing. Among these models there is the stochastic block model (SBM) proposed by Anderson et al. [1992] and Holland et al. [1983]. It is a probabilistic random graph model that aims to produce classes, called blocks, or more generally clusters in networks. This model has been used in several domains such as networks and biology sciences (Fortunato [2010], Porter et al. [2009]) as well as in statistics and machine learning (Goldenberg et al. [2010]). This model is a generalization of the Erdös-Réyni model proposed by Erdös ans Rényi [1960] using a latent structure on the nodes. In this model, the nodes of the network are grouped into disjoint blocks such that those belonging to the same block have the same probability of connection between them. In addition, all these nodes have the same probability of being connected to other nodes that belong to another block. The probability of existence of an edge between two nodes depends only on the blocks where the two nodes belong.

Mariadassou et al. [2010] have proposed a generalization of the SBM model to handle weighted graphs. Jernite et al. [2014] have treated the SBM model with categorical edges, Airoldi et al. [2008] and Latouche et al. [2011] have focused on the SBM model with superimposed clusters. More recently, Yang et al. [2011], Xu and Hero [2013], Zreik et al. [2017] and Matias and Miel [2017] have extended the SBM model to deal with dynamic networks and Barbillon et al. [2017] have dealt with the case of multiplex networks, where several edges can exist between a pair of nodes. These edges represent the different types of relationship between these nodes.

Several authors have focused on estimating parameters in the SBM model. First, Snijders and Nowicki [1997] have proposed a maximum likelihood inference based on the expectation maximization (EM) algorithm to estimate the probabilities of connection between nodes and to predict the clusters in the SBM model having only two blocks. Then, Nowicki and Snijders [2001] have generalized the previous work to treat the SBM model with an arbitrary number of blocks using a Bayesian approach based on Gibbs sampling. Since the EM algorithm requires the calculation of the distribution of Z conditionally on the observations X, which is generally intractable given that the edges of the network are not independent, Daudin et al. [2008] and Jaakola [2000] have introduced approximate methods based on a variational approach to estimate the parameters of the model and classify the nodes of the networks. They used the variational expectation maximization (VEM) algorithm. In addition, Latouche et al. [2012] have used a variational Bayesian inference based on a variational Bayesian expectation maximization algorithm (VBEM), whereas Nowicki and Snijders [2001] used the Gibbs sampling algorithm.

In most of the methods already discussed in this context, we emphasize that the SBM model is limited to binary networks, in which the edges are not weighted. Since most networks are weighted, Thomas and Blitzstein [2011] have proposed to apply a threshold to the weighted edges. This method is not efficient since it produces binary graphs in which only some of the relevant informations will be retained and the others will be lost. However, Mariadassou et al. [2010], Karrer and Newman [2011] and Ball et al. [2011] have dealt with the case of weighted SBM without thresholding. They have treated the SBM model with Poisson distributed weights.

Chapter 2 and 3 deal with the case of co-citation networks in a context of text mining. These networks are composed of words that represent the nodes of the network and of edges joining each pair of these words. Each edge is associated with an integer value representing the capacity or strength of links between a pair of words. These networks are then weighted according to the number of documents in the given corpus citing simultaneously this pair of words.

In chapter 2, we develop a SBM with binomial distributed weights. The binomial distribution takes the parameter m and the parameter $(\pi_{qr})_{q,r}$ as input. The parameter m is the maximum number of documents in the given corpus while the parameter $(\pi_{qr})_{q,r}$ is the probability matrix of connection between the two clusters q and r. Then, we use the variational expectation maximization (VEM) algorithm to estimate the parameters of the model as well as to classify the terms present in the documents of the corpus. We then adopt an integrated classification likelihood (ICL) criterion to select the optimal number of clusters. Finally, we introduce simulated data and then some real data to show the efficiency of our approach. We also compare our approach to the SBM model with Poisson distributed weights (PSBM).

In chapter 3, we treat the SBM model with binomial distributed weights using the variational Bayesian expectation maximization (VBEM) algorithm. This method allows us to estimate the parameters of the proposed model. In addition, we select the optimal number of clusters using the integrated likelihood variational Bayes (ILvb) criterion. We resume the same data introduced in chapter 2 to compare the results obtained using this approach with the results obtained using the VEM algorithm. Furthermore, we introduce an application to analyze a corpus of interviews with migrant minors, from Sub-Sahara to the European Mediterranean coast. These migrant minors have accepted to answer to a semi-directed interview. Their certificates have been put into numeric texts constituting the corpus. The observed network consists of 25 terms joined by edges. These terms are used in the certificates of the minors. Each edge joining a pair of terms is associated to a weight representing the number of interviews in which the two terms are used together. Finally, we compare the results obtained by applying the VBEM to those obtained by using the VEM.

In chapter 4, we treat the case of binary networks with a vector of weights associated to each node of this network. The objective of this chapter is to specify the different cognitive treatments performed by the brain during the preparation of handwriting from the electrical activity produced by neurons of the brain and recorded by the electroencephalogram. Furthermore, it aims to explore the evolution of the average intensity of clusters over time by classifying the 128 electrodes obtained by the electroencephalographic (EEG) recordings. We develop a SBM model to classify the nodes of the given network. This model takes two matrices as input data, one is the adjacency matrix and the other is the matrix of weights associated to the nodes. We develop a VEM algorithm to estimate the parameters of the proposed model as well as to classify the weighted vertices.

In chapter 5, we develop a general conclusion of the thesis then we give outlooks and present future research works and perspectives.

1.2 Classical Clustering Algorithms in General Framework

1.2.1 k-means clustering algorithm

The k-means clustering method proposed by MacQueen [1986] and Anderberg [1973] is the most popular method for cluster analysis. It is based on the decomposition and it is widely used in data mining field. It consists in partitioning a given dataset through a number of clusters K fixed in principle to create high similarity in the cluster, and low similarity between clusters. This algorithm works on unlabeled numerical data and will automatically group them into a fixed number K of

clusters.

Let $X = \{x_1, \ldots, x_n\}$ be the dataset containing n data points and $c = \{c_1, \ldots, c_K\}$ be the set of K centroids. Each centroid c_k, for $k = 1, \ldots, K$, represents the mean value of the kth cluster points.

The algorithm goes through three steps. The first step is the initialization of the algorithm. In this step, we choose randomly K data points that we consider the initial centroids c_k, for $k = 1, \ldots, K$, of the K latent clusters because we don't know yet where the true center of each cluster since they are latent. In the second step, we associate each point of the given dataset X to the nearest centroid. In the third step, we calculate the K new centroids of the K obtained clusters. The value of the new centroid is going to be the mean of all the data points in each cluster. It represents the barycenter of the cluster resulting from the previous step. We associate the same data points to their nearest K new centroids. We repeat the second and third step until the centroids stop moving which mean that the algorithm converge.

We can clearly notice that the purpose of the `k-means` algorithm is to minimize the total distance between the data points $x_i,\ i = 1, \ldots, n$ in each cluster k, for $k = 1, \ldots, K$, and its cluster center c_k which is equivalent to minimize an objective function representing the within-group sum-squared dispersion

$$J = \sum_{k=1}^{K} \sum_{i=1}^{n} Z_{ik} \|x_i - c_k\|^2,$$

where $\|x_i - c_k\|^2$ is the Euclidean distance and Z_{ik} is equal to 1 if the data point x_i is assigned to the centroid c_k and 0 otherwise. This is equivalent to saying that Z_{ik} is equal to 1 if $k = \arg\min_q \|x_i - c_q\|^2$ and 0 otherwise.

We set the gradient of J equal to zero to calculate the values of the centroids at each step

$$\nabla_{c_k} J = 2 \sum_{i=1}^{n} Z_{ik}(x_i - c_k) = 0.$$

Thus, we obtain

$$c_k = \frac{\sum_{i=1}^{n} Z_{ik}\ x_i}{\sum_{i=1}^{n} Z_{ik}}, \quad k = 1, \ldots, K.$$

Algorithm 1 develop the `k-means` algorithm. Unfortunately there is no determined method to find the optimal number of clusters. So, we try the algorithm with different values of K, we evaluate them then we choose the best value.

1.2.2 Hierarchical clustering algorithm

The hierarchical clustering developed by Rokach et al. [2005] also called hierarchical cluster analysis (`HCA`) is a method of clustering which aims at building a

Algorithm 1 k-means algorithm.

Initialization : Initialize randomly $c_k^{(0)}, k = 1, \ldots, K$.

1: Update k, the function J and the vector c_k iteratively
For $i \in 1 : n$ **do**
$k = \arg\min_q \|x_i - c_q\|^2$
$Z_{ik} \leftarrow 1$, $Z_{ik} \leftarrow 0 \ \forall q \neq k$
end for
$c_k = \frac{\sum_{i=1}^n Z_{ik}\ x_i}{\sum_{i=1}^n Z_{ik}}, \quad k = 1, \ldots, K$
$J = \sum_{k=1}^K \sum_{i=1}^n Z_{ik} \|x_i - c_k\|^2$

2: Repeat Step 1 until J converge

hierarchy of clusters. It is an alternative approach to `k-means` clustering for identifying groups in the dataset. It does not require us to pre-specify the number of clusters to be generated as is required by the `k-means` approach.

There are two types of hierarchical clustering, Agglomerative and Divisive :

— *Agglomerative method* : This method is also called bottom-up clustering method. We assign each observation to one cluster, then we compute the similarity (also called distance) between each of the clusters and combine the two most similar clusters into a new bigger cluster of nodes. We proceed recursively until all the observations are member of just one single big cluster. The result is a tree which can be plotted as a dendrogram.

— *Divisive method* : This method is also called top-down clustering method. It is an inverse order of the Agglomerative. We assign all the observations to one cluster, then split the cluster into two least similar clusters. We perform the split recursively on each group until we obtain a cluster for each observation.

The hierarchical clustering requires the measure of similarity (for Agglomerative method) and dissimilarity between clusters (for Divisive method). So it is required first to determine the proximity matrix which contains the distance between each pair of observations using a distance function (i.e. Euclidean distance, Manhattan distance, etc.). Then, the obtained matrix is updated to measure the distance between clusters using one of these three following methods :

— Single Linkage : The distance between two clusters is calculated as the shortest distance between two points in each cluster. It can be expressed as follows

$$L(q_1, q_2) = \min(\text{dist}(x_{iq_1}, x_{jq_2})), \tag{1.1}$$

where q_1 are q_2 are two clusters, x_{iq_1} and x_{jq_2} represent all the nodes in the cluster q_1 and q_2 respectively, and dist is a chosen distance function between the nodes.

— Complete Linkage : The distance between two clusters is calculated as the longest distance between two points in each cluster. It can be expressed as follows

$$L(q_1, q_2) = \max(\text{dist}(x_{iq_1}, x_{jq_2})), \tag{1.2}$$

where q_1 are q_2 are two clusters, x_{iq_1} and x_{jq_2} represent all the nodes in the cluster q_1 and q_2 respectively, and dist is a chosen distance function between the nodes.

— Average Linkage : the distance between two clusters is calculated as the average distance between each point in one cluster to every point in the other cluster. It can be expressed as follows

$$L(q_1, q_2) = \frac{1}{n_{q1} n_{q_2}} \sum_{i=1}^{n_{q_1}} \sum_{i=1}^{n_{q_2}} \text{dist}(x_{iq_1}, x_{jq_2}), \tag{1.3}$$

where q_1 are q_2 are two clusters, n_{q1} and n_{q_2} are the number of nodes in the clusters q_1 and q_2 respectively, x_{iq_1} and x_{jq_2} represent all the nodes in the cluster q_1 and q_2 respectively, and dist is a chosen distance function between the nodes.

We present in the following the Agglomerative hierarchical algorithm in a single linkage distance case.

Algorithm 2 Agglomerative hierarchical algorithm.

Initialization : Initialize a set of observations $X = \{x_1, \ldots, x_n\}$.

1: We assign each data points to a single cluster
For $i \in 1 : n$ **do**
$q_i = \{x_i\}$
end for
$C = \{q_1, \ldots, q_n\}$

2: **while** |C|>1 **do**
$(q_{min1}, q_{min2}) = L(q_r, q_l) = \min(\text{dist}(x_{iq_r}, x_{jq_l}))$ for all q_r and q_l in C.
$C \leftarrow C \setminus \{\{q_{min1}\}, \{q_{min2}\}\}$
$C \leftarrow C \cup \{q_{min1}, q_{min2}\}$
end while

1.2.3 Spectral clustering algorithm

The spectral clustering developed by Demmel [1999] is one of the most widely used techniques for exploratory data analysis. Its goal is to partition the data points into disjoint clusters such as the data points in the same cluster have a high similarity while the data points in different clusters have low similarity. This algorithm goes through three steps.

- Step 1 : Create a similarity graph between the n data points to cluster. We present two ways to construct a such graph :
  - ε-neighborhood graph : In such graph, each vertex is connected to other vertices falling inside a ball of radius ε, for a fixed real value ε .
  - k-nearest neighbor graph : In such graph, each vertex is connected to its k nearest neighbors, where k is a fixed integer number.
- Step 2 : Form the associated Laplacian matrix with the created similarity graph, then compute its first k eigenvectors to define a feature vector for each point.
- Step 3 : Apply the `k-means` algorithm on these vectors to split the data points into k clusters.

1.3 Community Detection in Networks

A community is a group of actors/nodes that have special ties because they have particular affinities, or have similar characteristics, or share interests. In a graph, a community represents a set of nodes that are strongly linked to each other, and weakly linked with nodes outside the community. Noting that the links joining pairs of nodes in the graph may be directed or undirected. For example, Airline route maps is a directed network, where vertices represent airports and there is a link between two vertices if there is a direct flight from one of the vertices to the other one. Another example of a directed network is Instagram or twitter followers, where vertices represent individuals and there is a link between two individuals if one of the individuals follows the other one. In this case, the other individual may not follow him back. However, an example of undirected network is telephone system, where the vertices are homes and links are the cables connecting homes. Another example of undirected network is mobile phone calls, where vertices are individuals and links are phone calls.

The partition of the network can be in disjoint groups. In this case, a node in the network can belong to a single group. Or, it can be in groups with overlapping. In this case, a node can belong to multiple groups.

1.3.1 Community detection with hierarchical clustering algorithms

In this case, the network data is represented by its adjacency matrix X and the observations are the n nodes of the network. The hierarchical clustering requires two decisions : The choice of a distance function measuring the distance $\text{dist}(i, j)$ between any two nodes, i and j, in the network and the definition of the distance $d(q, r)$ between any two clusters of nodes, q and r.

To define the distance $\text{dist}(i, j)$ between two nodes i and j of the network, we have several choices that access the topological structure of the network through the degrees of the nodes, the means of the rows in the adjacency matrix X and the number of neighbors that these nodes have in common. For undirected network, these distances are :

— Euclidean distance :

$$\text{dist}(i, j) = \sum_{m=1}^{n} (X_{im} - X_{jm})^2.$$

— Cosine similarity measure :

$$\text{dist}(i, j) = \frac{\sum_{m=1}^{n} X_{im} X_{mj}}{\sqrt{\sum_{m=1}^{n} X_{im}^2} \sqrt{\sum_{m=1}^{n} X_{jm}^2}}.$$

— Standard Pearson correlation coefficient :

$$\text{dist}(i, j) = \frac{cov(X_i, X_j)}{\sigma_i \sigma_j}.$$

Note that $cov(X_i, X_j)$ is the covariance of rows i and j in the adjacency matrix while σ_i and σ_j are the variance of rows i and j in the adjacency matrix respectively.

Now, for the choice of the distance $d(q, r)$ between two clusters q and r, we have three options : the single Linkage (1.1), the complete Linkage (1.2) and the average linkage (1.3). Note that the hierarchical clustering algorithm adapted to network is the same as in the general framework developed in the previous section where the observations are the nodes of the network. Since in the case of network data, we do not have the concept of distances, we have defined some distances $\text{dist}(i, j)$ between each paires of nodes i and j. Recall that for **Agglomerative clustering**, each node is initially assigned to its own cluster. Then two nearest clusters are merged into the same cluster. This process is repeated until only one cluster is left. This clustering algorithm constructs a hierarchy of clusters from the nodes of the network. However, for **Divisive clustering**, all the nodes of the network are initially placed in a single cluster, then cluster is subdivided into two least similar clusters. The split is performed recursively until each node forms a separate cluster of its own.

1.3.2 Community detection with `K-means` clustering algorithms

The `K-means` clustering algorithm adapted to network is the same as in the general framework developed in the previous section where the observations are the nodes of the given network and since in this case, we do not have concepts as distance or center, we define one based on the number of connecting edges of the nodes.

We define the ratio of the connecting edges of a node i and the size of the cluster q as the distance between the node i and the cluster q. In this case, we do not have to define a specific center of clusters, since the clusters in whole are going to serve as centers.

Recall that the algorithm iterate between three steps. The first step consists in assigning the nodes to K clusters randomly. The second step consists in calculating the distance to each cluster. This distance is defined by the number of edges connecting the node to the nodes of the cluster divided by the size of the cluster. The last step consists in reassigning the nodes to the nearest cluster based on the greatest distance.

1.3.3 Community detection with spectral clustering algorithms

In this case, the network data is represented by its adjacency matrix X and the observations are the n nodes of the network. The spectral clustering algorithm adapted to network is the same as in the general framework developed in the previous section where the observations are the nodes of the network. So we have the adjacency matrix X as input of the algorithm. Recall that the algorithm compute first the graph Laplacian L, then compute the K eigenvectors associated to the k smallest eigenvalues. At the end, the `k-means` algorithm is applied on these vectors to split the nodes into k clusters.

1.4 Networks Characteristics

This section aims to introduce some characteristics of the networks. First, we present some properties of nodes in the network, then we present some properties of edges in the networks. At the end, we present some global characteristics of the network.

1.4.1 Assortative and disassortative mixing

A network is said to show assortative mixing (also called homophily) if the nodes in the network that have many connections tend to be connected to other nodes with many connections and the nodes that have little connections tend to be connected to other nodes with little connections. This means that nodes with high degree in the network tend to be connected to the other higher degree nodes and the nodes with low degree tend to be connected to the other lower degree nodes. In social network, individuals have a strong tendency to connect with other individuals who are similar to them. For example, in friendship network, where individuals are connected to each others if there are at the same location. Noting that nodes with high degree form the core of network and the nodes with low degree lie on the boundary.

However, A network is said to show disassortative mixing (also called heterophily) if the nodes in the network that have many connections tend to be connected to other nodes with little connections and inversely. This means that nodes with high degree in the network tend to be connected to the other lower degree nodes and the nodes with low degree tend to be connected to the other higher degree nodes. In biological networks, individual have a strong tendency to connect with other individuals who are dissimilar to them. For example in dating networks, where large majority of edges are running between men and women. Figure 1.1 shows the difference between an assortative and a disassortative mixing network. This figure is available on https://stepsandleaps.wordpress.com/2013/08/15/.

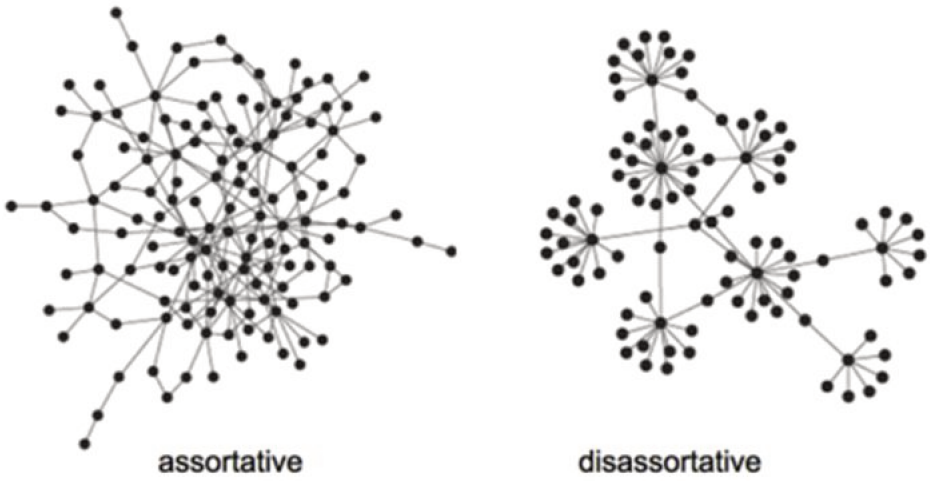

FIGURE 1.1 – Difference between an assortative and a disassortative mixing network.

1.4.2 Characteristics of nodes in networks

We present here some properties of nodes which determine the particularization of a network.

— *Degree Centrality* : It is a measure that count the number of neighbors of the node. In undirected graph, it is a measure of the number of edges connected to the node. However, in directed graph, we have two versions of the measure : in-degree which is the number of in-coming links and out-degree which is the number of out-going links. In this case, the degree centrality measure is a combination of the two measures.

— *Betweenness Centrality* : is a measure of centrality in a graph based on shortest path. Recall that shortest path is a path between vertices in a graph such that either the number of edges that the path passes through is minimum for unweighted graphs or the total sum of its constituent edges weights is minimum for weighted graphs. Betweenness centrality of a node measures the number of shortest paths that pass through the vertex.

— *Closeness Centrality* : Closeness centrality indicates how close a node is to all other nodes in the network. It is calculated as the average of the shortest path length from the node to every other node in the network. It highlights nodes that may reach any other nodes within a few hops and nodes that may be very distant in the graph.

— *Eigen Vector Centrality* : It is also called eigencentrality. It is a measure of the influence of a node in a network. It measures the importance of nodes for the connectivity of the network.

1.4.3 Characteristics of edges in networks

We present here some properties of edges (links) in the network.

— *Shortest path* : It is the path that connect two nodes with the shortest number of edges in unweighted graph. However, in weighted graph, it is the path that connect two nodes with the shortest sum of its edges weights. Recall that a path between two nodes is a any sequence of non-repeating nodes that connects the two nodes.

— *Geodesic Distance* : It is the shortest path between pair of nodes.

— *Diameter* : It is the longest shortest path between pairs of nodes. Or equivalently, the average distance between two randomly selected nodes.

— *Density* : It is the ratio of the number of edges in the network over the total number of possible edges between all pairs of nodes.

— *Triplet count* : It is the number of triangle formations in network.

1.4.4 Global characteristics in networks

We present here some global characteristics of the network.

— *Clustering Coefficient* : The number of closed triplets in the node's neighborhood over the total number of triplets in the neighborhood.

— *Component* : A component is a group of nodes that are all connected to each other, directly or indirectly. It is a connected subgraph where there is a path between every pair of vertices in this subgraph, but no vertex in the component can have an edge to another component.

— *Giant Component* : Is the component that is much bigger than every other component of the network.

1.5 Erdös-Rényi Random Graph Model

When analyzing a network, one of the approaches is to look at this network as a single fixed entity. But sometimes, it is useful to consider the edges as random variables. With this random network perspective, a given network is more than a single object. Instead, we can view the random network as a sample from a probability distribution. We can then study the whole probability distribution to gain insight into the network.

From a modeling perspective a network is a relatively simple object, consisting of only nodes and links. The real challenge, however, is to decide where to place the links between the nodes so that we reproduce the complexity of a real system. In this respect, the philosophy behind a random network is simple : We assume that this goal is best achieved by placing the links randomly between the nodes. That takes us to the definition of a random network.

The simplest and oldest network model is the random model, also known as Erdös-Rényi model. These network models have many properties in common with graphs encountered in the real world, and many properties that are very different.

According to this model, a network is generated by laying down a number n of nodes and adding edges between them with independent probability p for each node pair.

The Erdös-Rényi model developed by Erdös ans Rényi [1960] is one of the most important mathematical models for generating random graphs. A general random graph is defined by $G(n, m)$, where n is the number of vertices and m is the number of edges among those vertices chosen randomly. Since the graph has n vertices, so it can have up to $\binom{n}{2} = (n^2 - n)/2$ possible edges among these vertices. Therefore $m \leq (n^2 - n)/2$. We can then represent a graph by a vector X with $\binom{n}{2}$ entries which takes values in $\{0, 1\}$. Each entry represents a possible edge between two vertices in the graph. An entry with value 1 indicates that the

corresponding edge appears in the graph, while the value 0 indicates that the edge does not appear. So that, the vector $X \in \{0,1\}^{\binom{n}{2}}$. Therefore, the Erdös-Rényi model can be represented as a sequence of $\binom{n}{2}$ i.i.d random variables, where each one is a Bernoulli variable with success probability p.

We can define the model in an equivalent way by specifying the probability of observing each edge in the graph instead of specifying m edges. Therefore, the generation of the random graph goes as follows. We start with some number n of disconnected vertices. Then, we go over all possible edges one by one, and independently and each one with probability $0 \leq p \leq 1$. We have now a random graph of n vertices. It is then defined by $G(n,p)$, where p is the probability of interaction between each pair of node.

let X be the observed symmetric adjacency matrix encoding the interaction between nodes. So each variable X_{ij} between each pair of nodes i and j is defined as following

$$\begin{cases} X_{ij} = X_{ji} = 1 & \text{if } i \text{ and } j \text{ interact} \\ X_{ij} = 0 & \text{otherwise.} \end{cases}$$

We note by T_i the degree of the vertex i, for $i \in \{1,\dots,n\}$, defined by

$$T_i = \sum_{i \neq j} X_{ij},$$

which means the total number of neighbors of i.

The edges $X_{ij},\ i,j \in \{1\dots,n\}$ are independent and sampled from a Bernoulli distribution

$$X_{ij} \sim \mathbb{B}(p),$$

where p is the probability that an edge is present between i and j. So that

$$X_{ij} = \begin{cases} 1 & \text{with probability } \ p \\ 0 & \text{with probability } 1-p. \end{cases}$$

Since the edges X_{ij}, for $\{i,j\} \in \{1,\dots,n\} \times \{1,\dots,n\}$, are independent and identically distributed, we have

$$\sum_{i,j}^{n} X_{ij} \sim \mathbb{B}\left(\frac{n(n-1)}{2}, p\right).$$

Let E be the total number of edges in the graph. Then, we have

$$E = \mathbb{E}\left(\sum_{i,j}^{n} X_{ij}\right) = \frac{n(n-1)}{2} p,$$

where $\mathbb{E}$ denotes the expectation.

The degree T_i of each vertex i has a binomial distribution as follows

$$T_i \sim \mathbb{B}(n-1, p),$$

where $n-1$ is the maximal number of neighbors of i and p is the probability that the given node i has the degree T_i. Note that for graphs with a large number of nodes n, and for small value of the probability p, the Binomial distribution of the degree T_i is approximately a Poisson distribution as following

$$T_i \sim \mathbb{B}(n-1, p) \approx \mathbb{P}(\lambda),$$

where λ is the average node degree equal to $p(n-1)$.

Since the real world networks edge's are not independent, the Erdös-Rényi model poorly fits these networks. Thus, we define in the following the stochastic blockmodel (`SBM`) able to encode some more heterogeneity.

1.6 Stochastic Blockmodel for Binary Graphs

The stochastic block model (`SBM`) developed by Nowicki and Snijders [2001] is a random graph model generalizing the Erdös-Réyni model (Erdös ans Rényi [1960]) using a latent structure on the nodes. It is widely used as a canonical model to study clustering and community detection. This model aims to partition the vertices of a network into groups called blocks, or more generally clusters.

1.6.1 Notations and symbols of the model

In this section, we introduce some notations and symbols that are used while defining the stochastic block model.

— n : the total number of vertices in the network.
— Q : the fixed number of clusters in the networks.
— E : the total number of edges in the network.
— Z : a binary matrix of dimension $n \times Q$ where each cell Z_{iq} in the matrix indicates the group to which vertex belongs and can be expressed as follows $\forall i \in \{1, \ldots, n\}$,$\forall q \in \{1, \ldots, Q\}$,

$$Z_{iq} = \begin{cases} 1 & \text{if vertex } i \text{ belongs to cluster q} \\ 0 & \text{otherwise.} \end{cases}$$

The indicator variables $\{Z_{iq}\}$, for $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, Q\}$, are independent.

— α : a vector of length Q such that for all $q \in \{1, \ldots, Q\}$, α_q indicates the probability of belonging of a vertex to the cluster q. It can be expressed as follows : $\forall q \in \{1, \ldots, Q\}, \forall i \{1, \ldots, n\}$,

$$\alpha_q = \mathbb{P}\{Z_{iq} = 1\}.$$

Since each vertex in the graph belongs to only one cluster, we have $\sum_{q=1}^{Q} \alpha_q = 1$.

— X : the network represented by its adjacency matrix of dimensions $n \times n$ which encodes the observed interactions between the vertices of the network. An interaction between two vertices in the network is represented by an edge joining these two vertices. We have for all $i, j \in \{1, \ldots, n\}$,

$$X_{ij} = \begin{cases} 1 & \text{if node } i \text{ and node } j \text{ interact} \\ 0 & \text{otherwise.} \end{cases}$$

— π : a matrix of dimension $Q \times Q$ specifies the probability of interaction within the groups and outside the groups. These probabilities are noted by intra-group probability and inter-group probability respectively. For all $q, l \in \{1, \ldots, Q\}$, π_{ql} represents the probability of interaction between vertices belonging to cluster q and vertices belonging to cluster l such that : $\forall q, l \in \{1, \ldots, Q\}$,

$$\pi_{ql} = \mathbb{P}(X_{ij} | i \in \text{group } q, j \in \text{group } l).$$

Note that for undirected network, we have $\pi_{ql} = \pi_{lq}$.

1.6.2 Generation of the stochastic blockmodel data's

As we have already defined in the previous section, $Z = (Z_i)_{i \in \{1, \ldots, n\}}$ is a latent vector of $(\{0, 1\}^Q)^n$ describing the belonging of the node i to cluster q when $Z_{iq} = 1$ and not when $Z_{iq} = 0$. Since a node i can belong to only one cluster then we have $\sum_{q=1}^{Q} Z_{iq} = 1, \forall i$. The vectors Z_i for $i \in \{1, \ldots, n\}$ are independents and sampled from a multinomial distribution as follows

$$Z_i \sim \mathbb{M}(1, \alpha = (\alpha_1, \ldots, \alpha_Q)),$$

where $\alpha = (\alpha_1, \ldots, \alpha_Q)$ is the vector of class proportions defined in the previous section. We have

$$\sum_{q=1}^{Q} \alpha_q = 1.$$

Moreover, we suppose that the edges of the graph are conditionally independent given the label of the vertices i and j. Furthermore, we suppose that they are

sampled from a Bernoulli distribution as follows

$$X_{ij}|\{i \in q, j \in l\} \sim \mathcal{B}(\pi_{ql}),$$

where π is the $Q \times Q$ matrix of connection probabilities defined in the previous section.

Let $\theta = (\alpha, \pi)$. We are interested in the following in estimating the parameter θ and the latent variable Z in an undirected network without self loop. However, note that the obtained results can be extended to directed networks, with or without self-loops.

1.6.3 Maximum likelihood and log-likelihood

The Maximum Likelihood estimation (MLE) is a method that aims to estimate an optimal fit for the distribution of the data. This method provides an estimation of the parameters by finding the parameter values that maximize the likelihood function. The estimates are called maximum likelihood estimates. The computation of the likelihood equation is expensive and time consuming since this equation tend to become complex. Then, we think about simplifying this equation by using the log-likelihood equation instead of the likelihood equation. In fact, because the logarithm is monotonically increasing function of its argument, maximizing the log of a function is equivalent to maximizing the function itself. Taking the log not only simplifies the subsequent mathematical analysis, but it also helps numerically because the product of a large number of small probabilities can easily underflow the numerical precision of the computer, and this is resolved by computing instead the sum of the log probabilities.

1.6.4 Log-likelihood of the complete and incomplete data

In this model, the dataset is incomplete since there are some latent variables that influence the distribution of the data and the formation of the clusters within the network. Thus, we are interested in calculating the log-likelihood of the observed data (also called incomplete data).

We start first by calculating the log-likelihood of the complete data. We denote by $X = \{X_{ij}\}_{i,j=1,\dots,n}$ the set of all the edges in the graph and by $Z = \{Z_{iq}\}_{i=1,\dots,n}^{q=1,\dots,Q}$ the set of all the indicator variables. The joint distribution is defined by

$$\mathbb{P}_\theta(X, Z) = \mathbb{P}_\pi(X|Z)\mathbb{P}_\alpha(Z),$$

where

$$\begin{aligned}\mathbb{P}_\pi(X|Z) &= \prod_{i<j}^{n}\prod_{q,l}^{Q}\mathbb{P}_{\pi_{ql}}(X_{i,j}|Z_i,Z_j)\\ &= \prod_{i<j}^{n}\prod_{q,l}^{Q}\mathbb{P}_{\pi_{ql}}(X_{i,j})^{Z_{iq}Z_{jl}}\\ &= \prod_{i<j}^{n}\prod_{q,l}^{Q}(\pi_{ql}^{X_{ij}}(1-\pi_{ql})^{1-X_{ij}})^{Z_{iq}Z_{jl}}\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}_\alpha(Z) &= \prod_{i}^{n}\prod_{q}^{Q}\mathbb{P}_{\alpha_q}(Z_i)\\ &= \prod_{i}^{n}\prod_{q}^{Q}\alpha_q^{Z_{iq}}.\end{aligned}$$

Then, the log-likelihood of the complete data can be expressed as follows

$$\begin{aligned}\log\mathbb{P}_\theta(X,Z) &= \log\mathbb{P}_\alpha(Z)+\log\mathbb{P}_\pi(X|Z)\\ &= \sum_i\sum_q Z_{iq}\log(\alpha_q)+\sum_{i<j}\sum_{q,l}Z_{iq}Z_{jl}\log\mathbb{P}(X_{ij}|\pi_{ql})\\ &= \sum_i\sum_q Z_{iq}\log(\alpha_q)+\sum_{i<j}\sum_{q,l}Z_{iq}Z_{jl}(X_{ij}\log\pi_{ql}\\ &\quad +(1-X_{ij})\log(1-\pi_{ql})).\end{aligned}\tag{1.4}$$

1.6.5 Log-likelihood of the incomplete data

We are interested here in computing the log-likelihood of the incomplete data which can be obtained by the summation of the complete data likelihood over all the possible values of the latent variable Z.

Thus, the log-likelihood of the incomplete data can be expressed as follows

$$\begin{aligned}\log\mathbb{P}_\theta(X) &= \sum_Z\log\mathbb{P}_\theta(X,Z)\\ &= \sum_Z\left(\sum_i\sum_q Z_{iq}\log(\alpha_q)+\sum_{i<j}\sum_{q,l}Z_{iq}Z_{jl}(X_{ij}\log\pi_{ql}+(1-X_{ij})\log(1-\pi_{ql}))\right).\end{aligned}$$

This equation requires the summation over all possible values of the unobserved variable Z. Thus, it is intractable for networks having a large number of vertices. Then, we propose to use the expectation maximization (EM) algorithm developed by Dempster et al. [1977] and McLachlan and Krishnan [2007] to tackle this issue.

1.6.6 Expectation maximization algorithm

The Expectation Maximization algorithm is a way to find maximum-likelihood estimates for model parameters when data is incomplete, has missing data points, or has unobserved (hidden) latent variables. It is an iterative way to approximate the maximum likelihood function. It involves two steps in an iterative manner :

— Step 1 : this step is called the expectation step (`E`-step). In this step, we are interested in finding the expected value of the latent variables of the model by using the adjacency matrix associated to the observed network data and the current parameters of the model.

— Step 2 : this step is called the maximization step (`M`-step). In this step, we are interested in estimating the model parameters. So, we assume that the latent variables are equal to the current iteration estimate. Then, we maximize the expected log-likelihood found on the `E`-step with respect to the model parameters. These parameter-estimates are then used to determine the distribution of the latent variables in the next `E`-step.

The `EM` algorithm always improves an estimation of the parameters through this two-step process. However, it sometimes needs a few random starts to find the best model because the algorithm can hone in on a local maxima that isn't that close to the (optimal) global maxima.

1.6.7 Advantages and disadvantages of the `EM` algorithm

The `EM` algorithm has several advantages such as :

— The likelihood is guaranteed to increase for each iteration.

— The conceptual simplicity and the ease of implementation.

— It is guaranteed to converge to local optima.

However, it has several disadvantages such as :

— It can be very very slow, even on the fastest computer, due to large number of iterations involving high computation.

— It works well when the fraction of missing information is small and the dimensionality of the data is not too large. Indeed, the higher the dimensionality, the slower the `E`-step.

— It may converge to local maxima instead of converging to global maxima.

The `E`-step of the `EM` algorithm requires the computation of the conditional distribution of all the latent variables Z and model parameters, given the observed data X. This distribution can not be factorized and then intractable in the context of the `SBM` due to the dependency of the edges X_{ij} in the network. This, `EM` algorithm is no longer usable in this context because of the dependency structure on the observed edges.

In the following, we propose to use the variational expectation maximization (VEM) algorithm developed by Jordan et al. [1999] and Jaakkola and Jordan [2000].

1.6.8 Variational expectation maximization inference

The Variational Expectation Maximization (VEM) is an approximation maximization likelihood strategy based on variational approach.

We propose to rely on a variational decomposition. In the case of the SBM, it leads to

$$\log \mathbb{P}_\theta(X) = J_\theta(R_X(Z)) + \mathrm{KL}(R_X(Z) \parallel \mathbb{P}_\theta(Z|X)), \tag{1.5}$$

where $\mathbb{P}_\theta(Z|X)$ is the true conditional distribution of the latent variable Z given the observed variable X, $R_X(Z)$ is an approximate distribution of $\mathbb{P}_\theta(Z|X)$ and KL is the Kullback-Leibler divergence between $\mathbb{P}_\theta(Z|X)$ and $R_X(Z)$ defined by

$$\mathrm{KL}(R_X(Z) \parallel \mathbb{P}_\theta(Z|X)) = -\sum_Z R_X(Z) \log \frac{\mathbb{P}_\theta(Z|X)}{R_X(Z)}.$$

The KL measures the closeness of the two distributions $\mathbb{P}_\theta(Z|X)$ and $R_X(Z)$. Indeed, it helps us to measure how much information we lose by choosing $R_X(Z)$ as an approximation of $\mathbb{P}_\theta(Z|X)$.

Furthermore, since the Kullback-Leibler divergence is a non-negative measure. We have :

$$\mathrm{KL}(R_X(Z) \parallel \mathbb{P}_\theta(Z|X)) \geq 0. \tag{1.6}$$

We can underline that the equality is reached when $R_X(Z) = \mathbb{P}_\theta(Z|X)$.

However, $J_\theta(R_X(Z))$ is of the form

$$\begin{aligned} J_\theta(R_X(Z)) &= \sum_Z R_X(Z) \log \frac{\mathbb{P}_\theta(X,Z)}{R_X(Z)} \\ &= \sum_Z R_X(Z) \log \mathbb{P}_\theta(X,Z) - \sum_Z R_X(Z) \log R_X(Z) \\ &= \mathbb{E}_{R_X}[\log(\mathbb{P}_\theta(X,Z))] - \mathbb{E}_{R_X}[\log R_X(Z)], \end{aligned} \tag{1.7}$$

where $\mathbb{E}_{R_X}$ denotes the expectation with respect to distribution R_X.

The combination of the two equations (1.5) and (1.6) gives

$$\begin{aligned} \log \mathbb{P}_\theta(X) &= J_\theta(R_X(Z)) + \mathrm{KL}(R_X(Z) \parallel \mathbb{P}_\theta(Z|X)) \\ &\geq J_\theta(R_X(Z)). \end{aligned}$$

Therefore, $J_\theta(R_X(Z))$ is a lower bound of $\log \mathbb{P}_\theta(X)$.

By using the two equations (1.4) and (1.7), we obtain

$$\begin{aligned} J_\theta(R_X(Z)) &= \mathbb{E}_{R_X}[\log(\mathbb{P}_\theta(X,Z))] - \mathbb{E}_{R_X}[\log R_X(Z)] \\ &= \sum_i \sum_q \mathbb{E}_{R_X}(Z_{iq}) \log(\alpha_q) + \sum_{i<j} \sum_{q,l} \mathbb{E}_{R_X}(Z_{iq}Z_{jl})(X_{ij} \log \pi_{ql} \\ &\quad +(1-X_{ij}) \log(1-\pi_{ql})) + H(R_X), \end{aligned} \tag{1.8}$$

where $H(R_X)$ is of the form

$$H(R_X) = -\sum_i \sum_q \mathbb{E}_{R_X}(Z_{iq}) \log \mathbb{E}_{R_X}(Z_{iq}).$$

Note that, according to equality (1.7), optimizing the lower bound with respect to θ no longer requires the computation of the marginal likelihood. Furthermore, the equality $J_\theta(R_X(Z)) = \log \mathbb{P}_\theta(X)$ holds if and only if $R_X(Z) = \mathbb{P}_\theta(Z|X)$. Consequently, we can approximate $\mathbb{P}_\theta(Z|X)$ by $R_X(Z)$ in a certain class of distributions.

Now, we are interested in the maximization of the lower bound (1.8) with respect to the model parameters and the latent variable in the following class. Thus, we assume that the distribution $R_X(Z)$ can be factorized over the latent variable Z as follows

$$R_X(Z) = \prod_{i=1}^{n} R_{X,i}(Z_i) = \prod_{i=1}^{n} h(Z_i, \tau_i),$$

where τ_i is the variational parameter associated with Z_i such as $\sum_q^Q \tau_{iq} = 1$ for all $i \in \{1, \ldots, n\}$ and h is the multinomial distribution with the parameters τ_i. Thus, we have :

$$\tau_{iq} = \mathbb{P}(R_X(Z_{iq} = 1)) = \mathbb{E}(R_X(Z_{iq})) = \mathbb{E}_{R_X}(Z_{iq}) \tag{1.9}$$

and

$$\tau_{iq}\tau_{jl} = \mathbb{P}(R_X(Z_{iq} = 1, Z_{jl} = 1)) = \mathbb{E}(R_X(Z_{iq}, Z_{jl})) = \mathbb{E}_{R_X}(Z_{iq}, Z_{jl}). \tag{1.10}$$

Based on the equations (1.8), (1.9) and (1.10), the lower bound can be expressed of the form

$$\begin{aligned} J_\theta(R_X(Z)) &= \sum_i \sum_q \mathbb{E}_{R_X}(Z_{iq}) \log(\alpha_q) + \sum_{i<j} \sum_{q,l} \mathbb{E}_{R_X}(Z_{iq}Z_{jl})(X_{ij} \log \pi_{ql} \\ &\quad +(1-X_{ij}) \log(1-\pi_{ql})) + H(R_X) \\ &= \sum_i \sum_q \tau_{iq} \log(\alpha_q) + \sum_{i<j} \sum_{q,l} \tau_{iq}\tau_{jl}(X_{ij} \log \pi_{ql} + (1-X_{ij}) \log(1-\pi_{ql})) \\ &\quad - \sum_i \sum_q \tau_{iq} \log \tau_{iq}. \end{aligned} \tag{1.11}$$

1.6.9 Variational algorithm

In this section, we use the VEM algorithm to estimate the parameters of the model α and π and the latent variable Z. The VEM is an iterative method which involves the two steps :

— Step 1 : this step is called VE-step and aims to estimate the parameter τ. So we fix the model parameters α and π, then we maximize the lower bound (1.11) with respect to τ under the constraint $\sum_q^Q \tau_{iq} = 1, \forall i \in \{1, \ldots, n\}$.

— Step 2 : this step is called M-step and aims to estimate the model parameters α and π. So we fix the parameter τ, then we maximize the lower bound (1.11) with respect to the model parameters.

VE-step algorithm : The lower bound must be maximized with respect to τ under the constraint $\sum_q^Q \tau_{iq} = 1, \forall i \in \{1, \ldots, n\}$. As a consequence, using the Lagrange multiplier, we compute the derivative of $J_\theta(R_X(.)) + \lambda_i(\sum_q^Q \tau_{iq} - 1)$ with respect to τ_{iq}, for all $i \in \{1, \ldots, n\}$ and $q \in \{1, \ldots, Q\}$ and with respect to λ_i. Note that λ_i is the Lagrange multiplier.

According to (1.11), we have

$$\begin{aligned} J_\theta(R_X(Z)) + \lambda_i(\sum_q^Q \tau_{iq} - 1) &= \sum_{i<j}\sum_{q,l} \tau_{iq}\tau_{jl}(X_{ij}\log\pi_{ql} + (1-X_{ij})\log(1-\pi_{ql})) \\ &\quad - \sum_i \sum_q \tau_{iq}\log\tau_{iq} + \sum_i\sum_q \tau_{iq}\log\alpha_q \\ &\quad + \lambda_i(\sum_q \tau_{iq} - 1). \end{aligned} \tag{1.12}$$

By deriving (1.12) with respect to τ_{iq} and by taking this quantity equal to zero, we obtain :

$$\sum_l^Q \sum_{j=1, j\neq i}^n (X_{ij}\log\pi_{ql} + (1-X_{ij})\log(1-\pi_{ql}))\tau_{jl} + \log\alpha_q - \log\tau_{iq} - 1 + \lambda_i = 0.$$

Then, by deriving (1.12) with respect to λ_i and taking this quantity equal to zero, we obtain :

$$\sum_q^Q \tau_{iq} - 1 = 0.$$

This leads to the following fixed point relation

$$\begin{aligned} \hat{\tau}_{iq} &= e^{-1+\lambda_i}\alpha_q \prod_{j=1,j\neq i}^n \prod_l^Q \left(\pi_{ql}^{X_{ij}}(1-\pi_{ql})^{1-X_{ij}}\right)^{\hat{\tau}_{jl}} \quad \forall i \in \{1,\ldots,n\},\ \forall q \in \{1,\ldots,Q\} \\ &\propto \alpha_q \prod_{j=1,j\neq i}^n \prod_l^Q \left(\pi_{ql}^{X_{ij}}(1-\pi_{ql})^{1-X_{ij}}\right)^{\hat{\tau}_{jl}} \quad \forall i \in \{1,\ldots,n\},\ \forall q \in \{1,\ldots,Q\}, \end{aligned} \tag{1.13}$$

where $\propto$ means "proportional to" and $e^{(-1+\lambda_i)}$ is the normalizing constant. The equation (1.13) must be solved under the constraint $\sum_q^Q \tau_{iq} = 1$. The estimation of τ_{iq} is then obtained from (1.13) by iterating a fixed point algorithm until convergence. Note that τ need to be normalized after each iteration :

$$\hat{\tau}_{iq} = \frac{\hat{\tau}_{iq}}{\sum_{l=1}^{Q} \hat{\tau}_{il}}.$$

M-step algorithm : The lower bound must be maximized with respect to α and π. First, we fix the parameters τ and α, then we maximize the lower bound (1.11) with respect to π_{ql}. By deriving (1.11) with respect to π_{ql} and by taking this quantity equal to zero, we obtain :

$$\sum_{i<j} \tau_{iq}\tau_{jl} \left(\frac{X_{ij}}{\pi_{ql}} - \frac{(1 - X_{ij})}{(1 - \pi_{ql})} \right) = 0.$$

This leads to the following estimate of π_{ql}

$$\hat{\pi}_{ql} = \frac{\sum_{i<j} \tau_{iq}\tau_{jl} X_{ij}}{\sum_{i<j} \tau_{iq}\tau_{jl}}.$$

We fix now the parameters τ and π. The lower bound must be maximized with respect to α under the constraint $\sum_q^Q \tau_{iq} = 1, \forall i \in \{1, \ldots, n\}$. As a consequence, using the Lagrange multiplier, we compute the derivative of $J_\theta(R_X(.)) + \lambda_i(\sum_q^Q \alpha_q - 1)$ with respect to α_q, for all $q \in \{1, \ldots, Q\}$ and with respect to λ_i, for $i \in \{1, \ldots, n\}$. Recall that λ_i is the Lagrange multiplier.

According to (1.11), we have

$$\begin{aligned} J_\theta(R_X(Z)) + \lambda_i(\sum_q^Q \alpha_q - 1) \quad = \quad & \sum_{i<j} \sum_{q,l} \tau_{iq}\tau_{jl}(X_{ij} \log \pi_{ql} + (1 - X_{ij}) \log(1 - \pi_{ql})) \\ & + \sum_i \sum_q \tau_{iq} \log(\alpha_q) - \sum_i \sum_q \tau_{iq} \log \tau_{iq} \\ & + \lambda_i(\sum_q^Q \alpha_q - 1). \end{aligned} \tag{1.14}$$

By deriving (1.14) with respect to α_q and by taking this quantity equal to zero, we obtain :

$$\frac{1}{\alpha_q} \sum_i \tau_{iq} + \lambda_i = 0. \tag{1.15}$$

Then, By deriving (1.14) with respect to λ_i and by taking this quantity equal to zero, we obtain :

$$\sum_q \alpha_q - 1 = 0.$$

Since $\sum_q^Q \alpha_q = 1$, the equation (1.15) is the same by multiplying it by $\sum_q^Q \alpha_q$. So, multiplying (1.15) by α_q and summing over Q leads to

$$\lambda_i = -\sum_q \sum_i \tau_{iq}.$$

Thus, replacing λ_i by its value in (1.15) leads to

$$\hat{\alpha}_q = \frac{\sum_i \tau_{iq}}{\sum_q \sum_i \tau_{iq}}.$$

Moreover, since $\sum_q \tau_{iq} = 1$ then $\sum_q \sum_i \tau_{iq} = \sum_i \sum_q \tau_{iq} = n$. Thus, the estimation of α_q is equal to

$$\hat{\alpha}_q = \frac{1}{n} \sum_i \tau_{iq}.$$

1.6.10 Integrated complete data likelihood

We use the integrated classification likelihood (`ICL`) criterion in order to perform the selection of the most adequate number of blocks $\hat{Q}$. Roughly, this criterion is based on the complete data variational log-likelihood penalized by the number of parameters. It has been developed in a mixture context by Biernacki et al. [2000] and adapted to the stochastic block model by Daudin et al. [2008].

The `ICL` is of the form

$$\begin{aligned} ICL(Q) &= \sum_i \sum_q \hat{\tau}_{iq} \log \hat{\alpha}_q + \sum_{i<j} \sum_{q,l} \hat{\tau}_{iq} \hat{\tau}_{jl} (X_{ij} \log \hat{\pi}_{ql} + (1 - X_{ij}) \log(1 - \hat{\pi}_{ql})) \\ &\quad - \frac{V_Q}{2} \log n \\ &= \sum_i \sum_q \hat{\tau}_{iq} \log \hat{\alpha}_q + \sum_{i<j} \sum_{q,l} \hat{\tau}_{iq} \hat{\tau}_{jl} (X_{ij} \log \hat{\pi}_{ql} + (1 - X_{ij}) \log(1 - \hat{\pi}_{ql})) \\ &\quad - \frac{1}{2} \left(\frac{Q(Q+1)}{2} \log \frac{n(n-1)}{2} + (Q-1) \log n \right). \end{aligned}$$

where V_Q is the total number of parameters of the model for the Q clusters.

The `VEM` algorithm is run for different values of Q. The optimal number of clusters is chosen such that the `ICL` is maximized.

1.6.11 Initialization criteria of the algorithm

The implementation of the `VEM` algorithm raises two issues : the initialization of the algorithm and the convergence of the algorithm. We will discuss the issue related to the convergence of the algorithm in the next section. Now, for the initialization issue, the algorithm is run several times with different starting values, which are chosen by the k-means algorithm.

1.6.12 Stopping criteria of the algorithm

We already mentioned in the previous section that the implementation of the `VEM` algorithm raises the issue of the convergence of this algorithm. As our algorithm is an iterative procedure, we must test the convergence. A stopping criterion can be defined based on lower bound criterion $J_{\theta,\tau}$ or on the maximum number of iterations criterion as follows

— Lower bound criterion : We specify a threshold value ε. The algorithm cycles though the variational expectation step and the maximization step until the absolute distance between two successive values of the lower bound $J_{\theta,\tau}$ is smaller than the specified threshold value ε.

— Maximum iterations criterion : The algorithm stops running when it reaches the maximum number of iterations. This number is specified based on the size of the network.

Introduction en Français du Chapitre 2

Ceci est une introduction en français du chapitre "Estimation in a Binomial Stochastic Blockmodel for a Weighted Graph by a Variational Expectation Maximization Algorithm".

Introduction

Les modèles à blocs stochastiques ont été largement proposés en tant que modèles de graphe aléatoire probabiliste pour l'analyse des données ainsi que pour la détection des communautés dans les réseaux. Dans un certain nombre de réseaux du monde réel, les liens entre les noeuds n'ont pas tous le même poids. En fait, ils sont souvent associés à des poids qui les différencient en termes de force, d'intensité ou de capacité.

Nous avons déjà étudié dans le chapitre précédent, l'algorithme `SBM` pour traiter les réseaux non pondérés. Ces réseaux sont représentés par des graphes binaires où toutes les arêtes sont considérées comme identiques et non pondérées.

Cependant, dans ce chapitre, nous étudions le cas des réseaux pondérés, où chaque arête est associée à une valeur entière représentant sa capacité. Pour cela, nous fournissons un modèle `SBM` avec des arêtes pondérées distribuées selon une loi binomiale. Nous proposons une méthode d'inférence basée sur un algorithme espérance maximisation variationnel (`VEM`) afin de pouvoir estimer les paramètres dans ce modèle. Cette méthode est capable de traiter des réseaux fortement liés. Pour prouver la validité de cette méthode et mettre en évidence ses principales caractéristiques, nous introduisons certaines applications de l'approche proposée en utilisant des données simulées, puis un ensemble de données réelles. Nous comparons les clusters trouvés en utilisant notre approche avec les clusters trouvés en utilisant le modèle à blocs stochastiques avec des arêtes pondérées distribuées selon une loi de Poisson. Les résultats obtenus montrent que l'erreur statistique est plus faible pour le modèle à blocs stochastiques binomial que pour le modèle à blocs stochastiques avec des arêtes pondérées distribuées selon une loi de Poisson.

Motivation

La transformation numérique de la société défie les statistiques. Dans de nombreux contextes, la fouille de texte devient un outil standard utile pour trouver des modèles d'intérêt. C'est un intérêt croissant, en particulier pour les sciences sociales qui intègrent le numérique .

Au-delà des statistiques descriptives élémentaires et des modèles de comptage de mots, les réseaux de co-citations peuvent être facilement construits. Cela signifie que les données sont représentées par un graphe dont les noeuds sont des mots et les arêtes joignant chaque paire de mots sont pondérées en fonction du nombre de textes dans le corpus considéré citant simultanément cette paire de mots. La figure A.1 est un exemple d'un réseau de co-citation où les noeuds sont des articles sélectionnés parmi des articles très fréquemment cités (Ke et al. [2004]). De plus, ils sont publiés dans des revues scientifiques internationales. Les arêtes joignant chaque paire d'articles sont pondérées en fonction du nombre de co-citation de ces deux articles ensemble. Cette figure est disponible sur http://iv.slis.indiana.edu/ref/iv04contest. Dans cette figure, la largeur des arêtes joignant deux articles représente le nombre de co-citation des ces deux articles.

Une question d'intérêt général est de trouver des groupes de noeuds/mots plus étroitement liés. De nombreuses méthodes de détection de communautés ont été développées afin de s'attaquer à ce problème. Certains modèles de graphes aléatoires probabilistes comme le modèle d'Erdös-Renyi ou la famille stochastique blockmodel (`SBM`) peuvent être utilisés comme modèles paramétriques statistiques où les groupes inconnus sont des classes latentes. Au-delà du `SBM` binaire (dont les arêtes sont présentes/absentes), le modèle `SBM` avec une distribution plus générale de la valeur d'une arêtes pondérées joignant deux noeuds est d'un intérêt et d'une utilité croissants.

Dans ce chapitre, nous considérons le modèle `SBM` avec des arêtes pondérées distribuées selon une loi binomiale. Cette question est motivée par l'étude des réseaux de co-citations dans un contexte de fouille de textes où il y a un poids maximal m possible pour une arête correspondant au nombre de documents inclus dans le corpus. Outre l'élaboration et la mise en oeuvre de la procédure d'estimation dans le modèle `SBM` binomial, ce chapitre vise à comparer ce modèle avec le modèle `SBM` avec des arêtes pondérées distribuées selon une loi Poisson. En raison de la proximité bien connue entre les distributions binomiales et Poisson dans certains régimes de paramètres, est-ce que les procédures d'estimation pour ces deux modèles sont équivalentes ? Par exemple, un large corpus serait mieux modélisé par un modèle Poisson `SBM` ou bien par un modèle `SBM` binomial ? Quel est le nombre de clusters trouvés ? Comment est l'erreur statistique dans ces deux cas ? Suite à une procédure connue via un algorithme "espérance maximisation variationnel" (`VEM`) (Blei et al. [2017]), nous développons et nous mettons en oeuvre la méthode

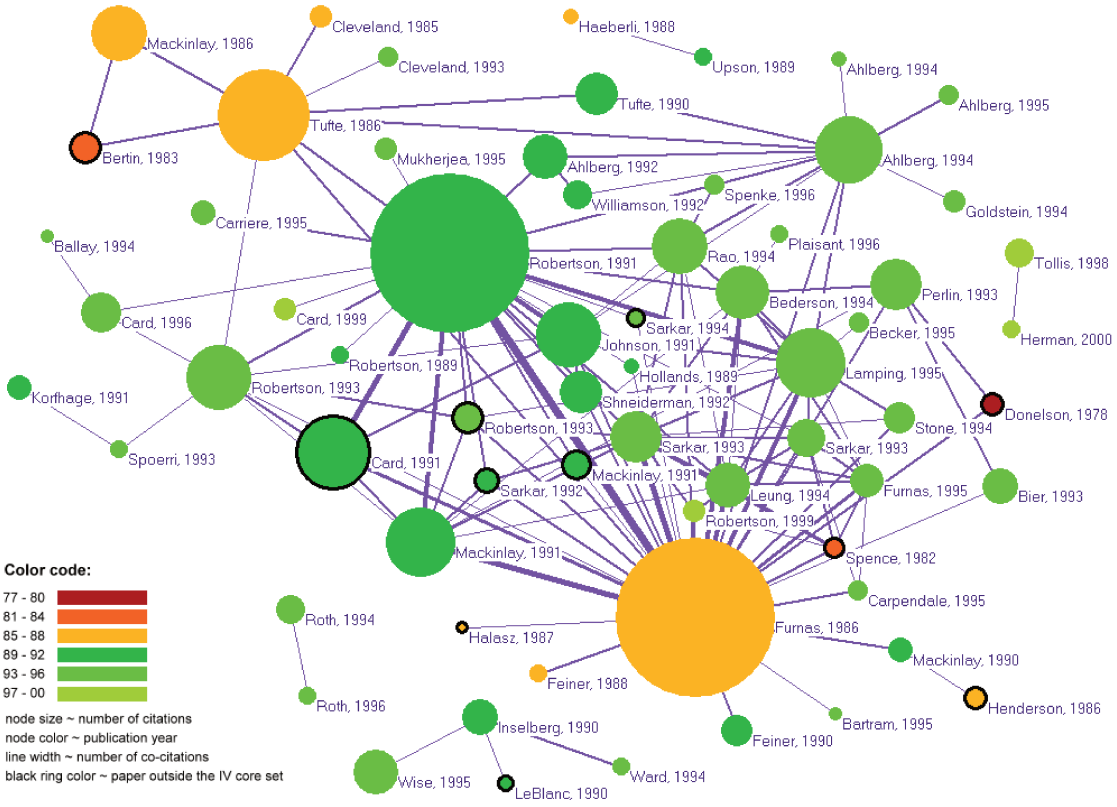

FIGURE A.1 – Réseau de co-citations d'articles fréquemment cités dans le domaine de la visulaisation des données entre 1974 et 2004.

sur des jeux de données simulées ainsi que sur un ensemble de données réelles : deux sur des réseaux de co-citations dans un contexte de fouille de textes ($m = 154$ et $m = 20$) et un dans un contexte de réseaux sociaux ($m = 14$).

Structure du Chapitre

Dans ce chapitre, nous définissons un réseau non orienté pondéré (poids valeurs entières bornées) et nous introduisons quelques notations dans la section 2.2. Nous définissons le modèle à blocs stochastiques avec des arêtes pondérées distribuées selon une loi binomiale dans la section 2.3. Dans la section 2.4, nous réalisons une inférence du modèle à blocs stochastiques binomial. En effet, dans la sous section 2.4.1, nous calculons la vraisemblance des données complètes tandis que dans la sous section 2.4.2, nous montrons que la vraisemblance des données incomplètes est intraitable. Puisque dans notre modèle, les données sont incomplètes, nous réalisons une inférence variationnelle dans la sous section 2.4.3 et puis nous introduisons l'algorithme de résolution dans la sous section 2.4.4. Dans la section

2.5, nous introduisons un critère de sélection du nombre optimal de clusters puis dans la section 2.6, nous appliquons notre méthode en introduisant des données simulées et des données réelles. D'autre part, dans la section 2.7, nous introduisons le modèle `SBM` avec des arêtes distribuées avec une loi de Poisson. Nous définissons ce modèle dans la sous section 2.7.1, puis nous calculons la vraisemblance des données complètes dans la sous section 2.7.2. Dans la sous section 2.7.3, nous réalisons une inférence variationnelle en utilisant l'algorithme `VEM` alors que dans la sous section 2.7.4, nous calculons le nombre optimal de clusters en introduisant un critère de sélection. Dans la dernière section 2.7.5, nous reprenons les données que nous avons déjà utilisées dans la section 2.6 afin d'appliquer le modèle `SBM` avec des arêtes distribuées selon une loi de Poisson et de comparer les résultats obtenus en utilisant cette méthode avec ceux obtenus en utilisant le modèle `SBM` avec des arêtes distribuées selon une loi binomiale.

Chapitre 2

Estimation in a Binomial Stochastic Blockmodel for a Weighted Graph by a Variational Expectation Maximization Algorithm

2.1 Introduction

Stochastic blockmodels have been widely proposed as probabilistic random graph models for data analysis as well as for the community detection in networks. In some real world networks, links between nodes do not all have the same weight. In fact, links between nodes are often associated with weights that differentiates them in terms of strength, intensity or capacity.

We have already studied in the previous chapter, the `SBM` algorithm for unweighted networks. These networks are represented by Binary graphs where all edges are considered to be identical and unweighted.

However, in this chapter, we study the case of weighted networks, where each edge is associated with an integer value representing the capacity of this link between the nodes. So we provide a `SBM` model with binomial distributed edges. We propose an inference method based on a variational expectation maximization (`VEM`) algorithm to estimate the parameters in the binomial stochastic block models. This method is able to handle large and strongly related networks. To prove the validity of this method and to highlight its main characteristics, we introduce some applications of the proposed approach using first simulated data, then using a set of real data. We compare clustering results using our approach to the cluste-

ring results using a stochastic block model with Poisson distributed weights. The obtained results show that the statistical error is lower for the binomial stochastic block model than for the stochastic block model with Poisson distributed weights.

Motivation

Digital transformation challenges statistics. In many contexts, text mining is becoming a standard useful tool to find patterns of interest. This is a rising interest in particular in digital humanities and social sciences.

Beyond elementary descriptive statistics and models counting words, co-citation networks may be easily built. It means data are represented by a graph whose nodes are words and edges between two words are weighted according to the number of texts in the considered corpus citing simultaneously this pair of words. Figure A.1 is an example of a co-citation network where nodes are papers and edges joining each pair of these papers are weighted according to the number of co-citation of these two papers together. These papers are selected from among very cited papers (Ke et al. [2004]). In addition, they are published in international scientific journals. In this figure, the width of the edges joining two papers represents the co-citation number of these two papers together.

A general question of interest is to find clusters of nodes/words more closely related. Lots of community detection methods were developed in order to tackle this issue. Some probabilistic random graph models like Erdös-Rényi or the stochastic blockmodel (`SBM`) family can be used as statistical parametric models where the unknown cluster are latent classes. Beyond binary `SBM` (whose edges are present/absent), `SBM` with a more general distribution for the value of an edge between nodes belonging to the same class are of increasing interest and usefulness.

In this chapter, we consider the `SBM` model with a binomial distribution on edges. This question is motivated by the study of co-citation networks in a text mining context where there is a maximal weight m possible for an edge corresponding to the number of documents included in the corpus. Beside developing and implementing the estimation procedure of a binomial SBM, this paper aims at comparing binomial `SBM` and `SBM` with a Poisson distributed weight. Due to the well known closeness between binomial and Poisson distributions in certain regimes of parameters, are the estimation procedures for these two models equivalent ? For instance, would be a large corpus be better modeled through a Poisson `SBM` or through a binomial one ? What is the number of clusters found ? How is the statistical error in these cases ? Following a known procedure through a variational Expectation Maximization (`VEM`) algorithm Blei et al. [2017], we develop and implement the method on simulated datasets (to validate the procedure) as well as benchmark real datasets : two in a co-citation text mining context ($m = 154$ and $m = 20$) and one in a social networks context ($m = 14$).

2.2 Specification and Notations of the Model

A general weighted undirected network is represented by $G := ([n], X)$, where $[n]$ is the set of nodes $\{1, ..., n\}$ for all $n \geq 1$ and X is the symmetric edge-weighted matrix of dimension $n \times n$ which encodes the observed interactions between nodes. We have for all $i, j \in \{1, \ldots, n\}$,

$$X_{ij} = \begin{cases} m_{ij} & \text{if the nodes } i \text{ and } j \text{ interact with an interaction strength } m_{ij} \\ 0 & \text{otherwise.} \end{cases}$$

We denote by E the total number of edges in the network. We assume that the nodes are not connected to themselves so that for all $i \in \{1, \ldots, n\}$, we have $X_{ii} = 0$. It means that all the diagonal elements of the weighted matrix X are equal to zero. The number of blocks in the graph is chosen equal to Q ($Q \geq 1$).

Let Z be a group membership indicator describing the belonging of the nodes to the clusters as follows : $\forall \{i, q\} \in \{1, \ldots, n\} \times \{1, \ldots, q\}$,

$$Z_{iq} = \begin{cases} 1 & \text{if the node } i \text{ belongs to cluster } q \\ 0 & \text{otherwise.} \end{cases}$$

Since a node i can belong to only one cluster, we have $\sum_{q=1}^{Q} Z_{iq} = 1, \forall i$. The matrix Z is of size $n \times Q$ and is composed of Z_{iq} for all $\{i, q\} \in \{1, \ldots, n\} \times \{1, \ldots, Q\}$.

2.3 Generation of the Stochastic Block Model data's

The vectors Z_i, for $i \in \{1, \ldots, n\}$, are independent and sampled from a multinomial distribution as following

$$Z_i \sim \mathcal{M}(1, \alpha = (\alpha_1, \ldots, \alpha_Q)),$$

where $\alpha = (\alpha_1, \ldots, \alpha_Q)$ is the vector of class proportions of length Q such as

$$\sum_{q=1}^{Q} \alpha_q = 1.$$

The variables $\{X_{ij}, i, j \in [n], i < j\}$ are independent conditionally on $\{Z_i = q, Z_j = l\}$, and are sampled from a binomial distribution as follows

$$X_{ij} | Z_{iq} Z_{jl} = 1 \sim \mathcal{B}(m, \pi_{ql}),$$

where

— m is the maximum weight associated to the edges.
— π is the $Q \times Q$ matrix of connection probabilities where π_{ql} represents the probability of existence of edge between the q-labeled and l-labeled nodes for all $q, l \in \{1, \ldots, Q\}$.

Then, the binomial stochastic blockmodel consists of the following parameters
— The latent variables $Z_i, \forall i \in \{1, \ldots, n\}$.
— The vector $\theta = (\alpha, \pi)$.

In the sequel, we are interested in estimating these parameters in a weighted undirected network without self loop. However, we affirm that all results obtained in this paper can be extended to directed networks, with or without self-loops.

2.4 Inference in the Binomial Stochastic Block Model

The dataset here is incomplete since there are some latent variables that influence the distribution of the data and the formation of the clusters within the network. We compute first the likelihood of the complete data, then we calculate the likelihood of the incomplete data. Furthermore, we develop an inference method to estimate the parameters of the model.

2.4.1 Likelihood of the complete data

We develop here the likelihood of the complete data. So, we define the joint distribution by

$$\mathbb{P}_\theta(X, Z) = \mathbb{P}_\pi(X|Z)\mathbb{P}_\alpha(Z),$$

where

$$\begin{aligned}
\mathbb{P}_\pi(X|Z) &= \prod_{i<j}^{n}\prod_{q,l}^{Q} \mathbb{P}_{\pi_{ql}}(X_{i,j}|Z_i, Z_j) \\
&= \prod_{i<j}^{n}\prod_{q,l}^{Q} \mathbb{P}_{\pi_{ql}}(X_{i,j})^{Z_{iq}Z_{jl}} \\
&= \prod_{i<j}^{n}\prod_{q,l}^{Q} \left(\binom{m}{X_{ij}} \pi_{ql}^{X_{ij}} (1-\pi_{ql})^{m-X_{ij}} \right)^{Z_{iq}Z_{jl}},
\end{aligned}$$

$$Z_{iq}Z_{jl} = \begin{cases} 1 & \text{if } i \text{ belongs to cluster } q \text{ and } j \text{ belongs to cluster } l \\ 0 & \text{otherwise.} \end{cases}$$

and

$$\begin{aligned}\mathbb{P}_\alpha(Z) &= \prod_i^n \prod_q^Q \mathbb{P}_{\alpha_q}(Z_i) \\ &= \prod_i^n \prod_q^Q \alpha_q^{Z_{iq}}. \end{aligned} \tag{2.1}$$

Thus, the log-likelihood of the complete data can be expressed as follows

$$\begin{aligned}\log \mathbb{P}_\theta(X, Z) &= \log \mathbb{P}_\pi(X|Z) + \log \mathbb{P}_\alpha(Z) \\ &= \sum_{i<j} \sum_{q,l} Z_{iq} Z_{jl} \log \mathbb{P}_{\pi_{ql}}(X_{ij}) + \sum_i \sum_q Z_{iq} \log(\alpha_q) \\ &= \sum_{i<j} \sum_{q,l} Z_{iq} Z_{jl} \left(\log \binom{m}{X_{ij}} + X_{ij} \log \pi_{ql} + (m - X_{ij}) \log(1 - \pi_{ql}) \right) \\ &\quad + \sum_i \sum_q Z_{iq} \log(\alpha_q). \end{aligned} \tag{2.2}$$

Note that possible values of the latent variable Z can be used to find the summation of complete data likelihood, which will determine the values of the log-likelihood of the incomplete data (given data) defined by X.

2.4.2 Likelihood of the incomplete data

The log-likelihood of the incomplete data can be expressed as follows

$$\log \mathbb{P}_\theta(X) = \log \sum_z \mathbb{P}_\theta(X, Z). \tag{2.3}$$

The equation above involves a summation over all the possible values of the latent variable Z. Thus may not be tractable except for small values of n, which means for small networks. To tackle this issue, we introduce the expectation maximization (`EM`) algorithm developed by Dempster et al. [1977] and McLachlan and Krishnan [2007]. This algorithm involves two steps in a iterative manner. The first step, called E-step, uses the current parameters of the model to determine the expected value of the latent variables of the model while the second step, called M-step, is used to maximize the log-likelihood (2.3), without calculating it, to estimate the parameters of the model assuming that the latent variables are fixed and equal to the current iteration estimate. However, the E-step is devoted to calculate the probability of the latent variables Z conditionally on the observed matrix X which is intractable in this context since the edges X_{ij} for $i, j \in \{1, \ldots, n\}$ are not independent. In fact, the edges X_{ij} between each two vertices i and j are marginally dependent and conditionally independent on the groups membership indicator of

the vertex i and the vertex j. Thus, the `EM` algorithm is no longer directly usable in this context because of the dependency structure on the observed edges.

We use in the following the variational expectation maximization (`VEM`) algorithm developed by Jordan et al. [1999] and Jaakkola and Jordan [2000] which is an approximation maximization likelihood strategy based on variational approach (Daudin et al. [2008]). This method overcomes the issue due to the mean field approximation which ensures that the latent variables Z_i are independent to each other, given the observed data X.

2.4.3 Variational inference

In the presence of the latent variables Z_i, we turn to the expectation maximization algorithm. However, this algorithm requires the evaluation of the conditional expectation $\mathbb{E}_{Z|X}[\log \mathbb{P}_\theta(X, Z)]$ which is intractable in this context since the latent variables Z_i depend conditionally on the observed matrix X. The variational approach avoid this limitation by maximizing a lower bound of the log-likelihood (2.3) based on an approximation of the true conditional distribution of Z given Y.

We rely on a variational decomposition of the incomplete log-likelihood (2.3) as following

$$\log \mathbb{P}_\theta(X) = J_\theta(R_X(Z)) + \mathrm{KL}(R_X(Z) \parallel \mathbb{P}_\theta(Z|X)), \tag{2.4}$$

where $\mathbb{P}_\theta(Z|X)$ is the true conditional distribution of Z given X, $R_X(Z)$ is an approximate distribution of $\mathbb{P}_\theta(Z|X)$ and KL is the Kullback-Leibler divergence between $\mathbb{P}_\theta(Z|X)$ and $R_X(Z)$ defined by

$$\mathrm{KL}(R_X(Z) \parallel \mathbb{P}_\theta(Z|X)) = -\sum_Z R_X(Z) \log \frac{\mathbb{P}_\theta(Z|X)}{R_X(Z)}.$$

It measures the closeness of the two distributions $\mathbb{P}_\theta(Z|X)$ and $R_X(Z)$. Furthermore, it is a non-negative measure :

$$\mathrm{KL}(R_X(Z) \parallel \mathbb{P}_\theta(Z|X)) \geq 0. \tag{2.5}$$

We can underline that the equality is reached when $R_X(Z) = \mathbb{P}_\theta(Z|X)$.

The term $J_\theta(R_X(Z))$ of the equation (2.4) is of the form

$$\begin{aligned} J_\theta(R_X(Z)) &= \sum_Z R_X(Z) \log \frac{\mathbb{P}_\theta(X, Z)}{R_X(Z)} \\ &= \sum_Z R_X(Z) \log \mathbb{P}_\theta(X, Z) - \sum_Z R_X(Z) \log R_X(Z) \\ &= \mathbb{E}_{R_X}[\log(\mathbb{P}_\theta(X, Z))] - \mathbb{E}_{R_X}[\log R_X(Z)], \end{aligned} \tag{2.6}$$

where $\mathbb{E}_{R_X}$ denotes the expectation with respect to distribution R_X.

The combination of (2.4) and (2.5) ensure that

$$\log \mathbb{P}_\theta(X) \geq J_\theta(R_X).$$

Therefore, $J_\theta(R_X)$ is a lower bound of $\log \mathbb{P}_\theta(X)$.

Moreover, $\mathbb{P}_\theta(Z|X)$ is not tractable because of the dependency of the variables X_{ij}. Thus, the classical property of KL which states that the lower bound $J_\theta(R_X)$ has a unique maximum $\mathbb{P}_\theta(X)$ reached for $R_X(Z) = \mathbb{P}_\theta(Z|X)$ is not helpful. So, we maximize $J_\theta(R_X)$ with respect to R_X and θ. By using the equations (2.6) and the log-likelihood of the complete data equation (2.2), the lower bound $J_\theta(R_X)$ can be written as follows

$$\begin{aligned} J_\theta(R_X) &= H(R_X) + \mathbb{E}_{R_X}[\log(\mathbb{P}_\theta(X,Z))] \\ &= H(R_X) + \sum_i \sum_q \mathbb{E}_{R_X}(Z_{iq}) \log \alpha_q + \sum_{i<j} \sum_{q,l} \mathbb{E}_{R_X}(Z_{iq}, Z_{jl})(\log \binom{m}{X_{ij}} \\ &\quad + X_{ij} \log \pi_{ql} + (m - X_{ij}) \log(1 - \pi_{ql})), \end{aligned} \tag{2.7}$$

where $H(R_X) = -\sum_i \sum_q \mathbb{E}_{R_X}(Z_{iq}) \log \mathbb{E}_{R_X}(Z_{iq})$.

The E-step of the `EM` algorithm becomes tractable when we assume that the distribution $R_X(Z)$ can be factorized over the latent variable Z as follows

$$R_X(Z) = \prod_{i=1}^{n} R_{X,i}(Z_i) = \prod_{i=1}^{n} h(Z_i; \tau_i), \tag{2.8}$$

where $\{\tau_i \in [0,1]^Q, i = 1, \ldots, n\}$ are the variational parameters associated with $\{Z_i, i = 1, \ldots, n\}$ such as $\sum_q \tau_{iq} = 1, \ \forall i \in \{1, \ldots, n\}$ and h is the multinomial distribution with parameters τ_i. We have

$$\tau_{iq} = \mathbb{P}(R_X(Z_{iq} = 1)) = \mathbb{E}(R_X(Z_{iq})) = \mathbb{E}_{R_X}(Z_{iq}) \tag{2.9}$$

and

$$\tau_{iq}\tau_{jl} = \mathbb{P}(R_X(Z_{iq} = 1, Z_{jl} = 1)) = \mathbb{E}(R_X(Z_{iq}, Z_{jl})) = \mathbb{E}_{R_X}(Z_{iq}, Z_{jl}). \tag{2.10}$$

By using (2.8), (2.9), (2.10) and by developing the equation (2.7), we obtain that $J_\theta(R_X)$ can be written as follows

$$\begin{aligned} J_\theta(R_X) &= -\sum_i \sum_q \tau_{iq} \log \tau_{iq} + \sum_i \sum_q \tau_{iq} \log \alpha_q + \sum_{i<j} \sum_{q,l} \tau_{iq}\tau_{jl}(\log \binom{m}{X_{ij}} + X_{ij} \log \pi_{ql} \\ &\quad + (m - X_{ij}) \log(1 - \pi_{ql})). \end{aligned} \tag{2.11}$$

The estimation of the parameters θ and τ of the model requires the following two steps :

— Step 1 : we fix the parameter θ then we calculate τ by maximizing $J_\theta(R_X)$.
— Step 2 : we fix the parameter τ and we calculate the parameter $\theta = (\alpha, \pi)$ by maximizing $J_\theta(R_X)$.

VE-step algorithm. By fixing the parameter θ and by maximizing the lower bound $J_\theta(R_X)$ with respect to τ and under the condition $\sum_q \tau_{iq} = 1$, $\forall i \in \{1, \dots, n\}$, we can obtain $\hat{\tau}$ by the following fixed point relation

$$\hat{\tau}_{iq} \propto \alpha_q \prod_j \prod_l \left(\binom{m}{X_{ij}} \pi_{ql}^{X_{ij}} (1 - \pi_{ql})^{m - X_{ij}} \right)^{\hat{\tau}_{jl}}. \tag{2.12}$$

The estimation of τ is obtained from (2.12) by iterating a fixed point algorithm until convergence.

Proof. The lower bound must be maximized with respect to τ under the constraint $\sum_q^Q \tau_{iq} = 1, \forall i \in \{1, \dots, n\}$. As a consequence, using the Lagrange multiplier, we compute the derivative of $J_\theta(R_X(Z)) + \lambda_i(\sum_q^Q \tau_{iq} - 1)$ with respect to τ_{iq} for all $i \in \{1, \dots, n\}$, $q \in \{1, \dots, Q\}$ and λ_i. Note that λ_i is the Lagrange multiplier. According to (2.11), we have

$$\begin{aligned} J_\theta(R_X(Z)) + \lambda_i(\sum_q^Q \tau_{iq} - 1) = & \sum_i \sum_q \tau_{iq} \log(\alpha_q) + \sum_{i<j} \sum_{q,l} \tau_{iq}\tau_{jl} (\log \binom{m}{X_{ij}} + X_{ij} \log \pi_{ql} \\ & + (m - X_{ij}) \log(1 - \pi_{ql})) - \sum_i \sum_q \tau_{iq} \log \tau_{iq} \\ & + \lambda_i (\sum_q \tau_{iq} - 1). \end{aligned} \tag{2.13}$$

By deriving (2.13) with respect to τ_{iq} and by taking this quantity equal to zero, we obtain :

$$\sum_l^Q \sum_{j=1, j\neq i}^n (\log \binom{m}{X_{ij}} + X_{ij} \log \pi_{ql} + (m - X_{ij}) \log(1 - \pi_{ql}))\tau_{jl} + \log(\alpha_q) - \log \tau_{iq} - 1 + \lambda_i = 0.$$

Then, by deriving (2.13) with respect to λ_i and by taking this quantity equal to zero, we obtain :

$$\sum_q^Q \tau_{iq} - 1 = 0.$$

This leads to the following fixed point relation

$$\begin{aligned} \hat{\tau}_{iq} &= e^{-1+\lambda_i} \alpha_q \prod_{j=1, j\neq i}^n \prod_l^Q \left(\binom{m}{X_{ij}} \pi_{ql}^{X_{ij}} (1 - \pi_{ql})^{m - X_{ij}} \right)^{\hat{\tau}_{jl}} \quad \forall i \in \{1, \dots, n\}, \forall q \in \{1, \dots, Q\} \\ &\propto \alpha_q \prod_j \prod_l \left(\binom{m}{X_{ij}} \pi_{ql}^{X_{ij}} (1 - \pi_{ql})^{m - X_{ij}} \right)^{\hat{\tau}_{jl}} \quad \forall i \in \{1, \dots, n\}, \forall q \in \{1, \dots, Q\}, \end{aligned} \tag{2.14}$$

where $\propto$ means "proportional to" and $e^{(-1+\lambda_i)}$ is the normalizing constant. The equation (2.14) must be solved under the constraint $\sum_q^Q \tau_{iq} = 1$. The estimation of τ_{iq} is then obtained from (2.14) by iterating a fixed point algorithm until convergence. Note that the value of τ need to be normalized after each iteration :

$$\hat{\tau}_{iq} = \frac{\hat{\tau}_{iq}}{\sum_{l=1}^{Q} \hat{\tau}_{il}}.$$

□

M-step algorithm. We are interested here in the estimation of the parameters α and π. By fixing the parameter τ and by maximizing the lower bound $J_\theta(R_X)$ defined above with respect to α and under the condition $\sum_q \alpha_q = 1$, we obtain the following estimation of α_q

$$\hat{\alpha}_q = \frac{1}{n} \sum_i \tau_{iq}.$$

The proof is given in the previous chapter.

Then by maximizing the lower bound $J_\theta(R_X)$ with respect to π, we obtain the following estimation of π_{ql}

$$\hat{\pi}_{ql} = \frac{\sum_{i<j} \tau_{iq}\tau_{jl}X_{ij}}{m \sum_{i<j} \tau_{iq}\tau_{jl}}.$$

Proof. The lower bound must be maximized with respect to π. We fix the parameters τ and α, then we maximize the lower bound (2.11) with respect to π_{ql}. By deriving (2.11) with respect to π_{ql} and by taking this quantity equal to zero, we obtain :

$$\sum_{i<j} \tau_{iq}\tau_{jl} \left(\frac{X_{ij}}{\pi_{ql}} - \frac{(m - X_{ij})}{(1 - \pi_{ql})} \right) = 0.$$

This leads to the following estimate of π_{ql}

$$\hat{\pi}_{ql} = \frac{\sum_{i<j} \tau_{iq}\tau_{jl}X_{ij}}{m \sum_{i<j} \tau_{iq}\tau_{jl}}.$$

□

2.4.4 Algorithm of resolution

We denote by t the current index for iterations in the algorithm and by ε a fixed threshold of convergence.

Algorithm 3 Variational Expectation Maximization algorithm for inference in `SBM`

Initialization : Initialize τ^0 with a hierarchical algorithm based on the classical Ward distance by considering the Euclidean distance defined by $\text{dist}(i,j) = \sum_{m=1}^{n}(X_{im} - X_{jm})^2$.

1: Update the parameters τ and θ iteratively

$$\theta^{(t+1)} = \arg\max_{\theta} J_{\theta}(R_X; \tau^{(t)})$$

$$\tau^{(t+1)} = \arg\max_{\tau} J_{\theta^{(t+1)}}(R_X; \tau)$$

2: Repeat Step 1 until $\|\theta^{(t+1)} - \theta^{(t)}\| < \varepsilon$.

2.5 Integrated Classification Likelihood (ICL)

In the sections above, we estimated the parameters of the model by fixing the number of blocks Q since the `SBM` model function requires the number of latent groups Q as an input argument. We are interested here in choosing the number of clusters $\hat{Q}$ that will optimally fit the data. One of the proposed method consists in iterating the `SBM` model with different values of Q and then choosing the optimal number of clusters by evaluating goodness of fit for each group sizes (Lei [2016]). This method is expensive in terms of time and computing since we evaluate the goodness of fit for all the groups.

Another approach consists in using the Bayesian information criterion (BIC). The optimal number of clusters is obtained by running the model for different values of Q and then by choosing the one which provides the higher value of `BIC`. We have

$$BIC(Q) = \log \mathbb{P}_{\theta}(X) - \frac{V_Q}{2} \log n,$$

where V_Q is the number of parameters of the model for the Q groups. However, This method involves the computation of the log-likelihood of the given data X which is intractable.

Thus, Daudin et al. [2008] proposed the integrated classification likelihood (ICL) criterion to estimate Q in a `SBM` model. This method is an approximation of the complete data likelihood :

$$ICL(Q) \approx \mathbb{P}_{\theta}(X, Z|Q).$$

The ICL is of the form

$$\begin{aligned} ICL(Q) &= \sum_{i<j}\sum_{q,l} \hat{\tau}_{iq}\hat{\tau}_{jl}\left(\log\binom{m}{X_{ij}} + X_{ij}\log\hat{\pi}_{ql} + (m - X_{ij})\log(1-\hat{\pi}_{ql})\right) \\ &\quad + \sum_{i}\sum_{q} \hat{\tau}_{iq}\log\hat{\alpha}_q - \frac{V_Q}{2}\log n \\ &= \sum_{i<j}\sum_{q,l} \hat{\tau}_{iq}\hat{\tau}_{jl}\left(\log\binom{m}{X_{ij}} + X_{ij}\log\hat{\pi}_{ql} + (m - X_{ij})\log(1-\hat{\pi}_{ql})\right) \\ &\quad + \sum_{i}\sum_{q} \hat{\tau}_{iq}\log\hat{\alpha}_q - \frac{1}{2}\left(\frac{Q(Q+1)}{2}\log\frac{n(n-1)}{2} + (Q-1)\log n\right). \end{aligned}$$

The `VEM` algorithm is run for different values of Q and $\hat{Q}$ is chosen such that ICL is maximized.

2.6 Numerical experiments

This section aims at highlighting the main features of the proposed inference algorithm and to prove its validity by considering some simulated data and then applying our algorithm to a set of real data.

2.6.1 Simulated data

First, we perform the stochastic blockmodel using simulated data with a binomial output distribution. The graph has $n = 20$ vertices. We choose for this simulation a fixed number of clusters Q equal to three. We use in the simulation the following parameters :

$$\bar{\alpha} = (0.2, 0.5, 0.3)$$

and

$$\bar{\pi} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}.$$

We visualize the graph in Figure 2.1 using `Gephi` software with the layout algorithm "Force Atlas". We can show in Figure 2.1 the structure of the simulated data graph. There are three apparent communities.

Note that in all the graphs below, the width of the lines used to represent the edges is proportional to their weights, so that a line with a large width between two vertices indicates a strong relationship between these two vertices.

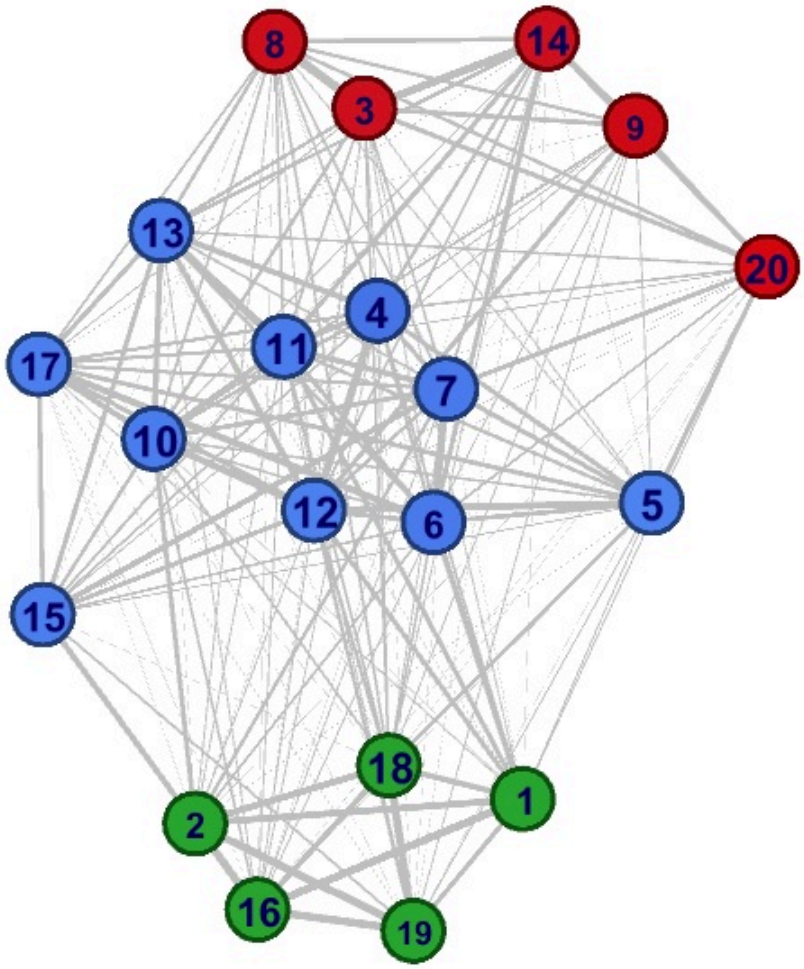

FIGURE 2.1 – First simulated data graph visualization with `Gephi`.

By applying our algorithm implemented in `R` programming language, we obtain that the vertices of the network are grouped into three clusters as shown in Table 2.1.

Clusters	**Vertices**
Red	3 8 9 14 20
Blue	4 5 6 7 10 11 12 13 15 17
Green	1 2 16 18 19

TABLE 2.1 – Grouping first simulated graph vertices into clusters

Table 2.1 shows clearly that the nodes of the first simulated graph are split into three clusters which are the same as the three clusters shown in the Figure 2.1. That confirms the effectiveness of our method. The time of convergence of the algorithm is 0.22 second (CPU Corel3 - 4GB RAM) which is so satisfying.

Now, we sample $S = 100$ random graphs according to the same mixture model, then we calculate in Table 2.2 and Table 2.3, for each parameter, the estimated Root Mean Squares Error (`RMSE`) defined by :

$$RMSE(\bar{\alpha}_q) = \sqrt{\frac{1}{S}\sum_{s=1}^{S}(\hat{\alpha}_q^{(s)} - \bar{\alpha}_q)^2}$$

and

$$RMSE(\bar{\pi}_{qr}) = \sqrt{\frac{1}{S}\sum_{s=1}^{S}(\hat{\pi}_{qr}^{(s)} - \bar{\pi}_{qr})^2},$$

where the superscript s labels the estimates obtained in simulation s. Note that we sort the estimated parameters $\hat{\alpha}_q$ in descending order to outcome the identifiability problem of the clusters.

RMSE$(\bar{\alpha}_1)$	**RMSE**$(\bar{\alpha}_2)$	**RMSE**$(\bar{\alpha}_3)$
0.011	0.003	0.004

TABLE 2.2 – Root Mean Squares Error of the parameter $\bar{\alpha}_q$ for the first simulated data using the binomial `SBM` model.

Table 2.2 shows that the `RMSE` of the parameters $\bar{\alpha}_q$, for $q \in \{1, 2, 3\}$, are close to zero, which means that the obtained estimated parameter $\hat{\alpha}$ is close to the observed parameter $\bar{\alpha}$.

RMSE	$\bar{\pi}_{.1}$	$\bar{\pi}_{.2}$	$\bar{\pi}_{.3}$
$\bar{\pi}_{1.}$	0.04	0.04	0.09
$\bar{\pi}_{2.}$	0.04	0.08	0.07
$\bar{\pi}_{3.}$	0.09	0.07	0.09

TABLE 2.3 – Root Mean Squares Error of the parameter $\bar{\pi}_{qr}$ for the first simulated data using the binomial `SBM` model.

Table 2.3 shows that the `RMSE` of the parameters $\bar{\pi}_{ql}$, for $\{q, l\} \in \{1, 2, 3\} \times \{1, 2, 3\}$, are close to zero, which means that the obtained estimated parameter $\hat{\pi}$ is close to the observed parameter $\bar{\pi}$.

In order to compare the estimated clustering results to the simulated ones, we propose to calculate the Adjusted Rand Index (ARI) proposed by Hubert and Arabie [1985]. It is defined as a measure of the similarity between two data clustering that lies between 0 (the two clusterings are completely independent) and 1 (identical clusterings). Note that a larger ARI means a high agreement between two partitions.

The average of the ARI between the simulated clustering results and the estimated clustering results obtained by using the proposed method is equal to 0.85. This means a high agreement between the two partitions of the nodes.

Now, we introduce a graph with a larger number of vertices to confirm the validity of the proposed algorithm for larger weighted networks. This graph has $n = 70$ vertices and a fixed number of clusters Q equal to five. The parameters used are :

$$\bar{\alpha} = (0.2, 0.1, 0.3, 0.35, 0.05)$$

and

$$\bar{\pi} = \begin{bmatrix} 0.5 & 0.1 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.4 & 0.2 & 0.1 & 0.2 \\ 0.1 & 0.2 & 0.6 & 0.05 & 0.05 \\ 0.1 & 0.1 & 0.05 & 0.4 & 0.35 \\ 0.2 & 0.2 & 0.05 & 0.35 & 0.2 \end{bmatrix}.$$

We visualize in Figure 2.2 the network's graph using Gephi software with the layout algorithm Force Atlas.

The structure of network graph in Figure 2.2 shows clearly five apparent communities. By applying our algorithm implemented in the software R, we obtain that the optimal number of clusters is five and that the nodes are grouped into these five clusters as shown in Table 2.4.

Clusters	Vertices
Cyan	1 12 13 14 17 20 23 31 33 45 47 49 50 51 52 57 68
Gray	3 5 6 9 11 21 22 25 29 32 34 37 38 39 42 44 46 58 62 64 65 66 67 69 70
Green	2 4 8 10 18 28 30 36 41 53 54 55 60 63
Blue	15 16 19 35 61
Red	7 24 26 27 40 43 48 56 59

TABLE 2.4 – Grouping second simulated graph vertices into clusters.

The nodes of the graph are grouped into the same five clusters shown in Figure 2.2.

We calculate in Table 2.5 and Table 2.6 the RMSE of the parameters $\bar{\alpha}_q$ and $\bar{\pi}_{qr}$ respectively.

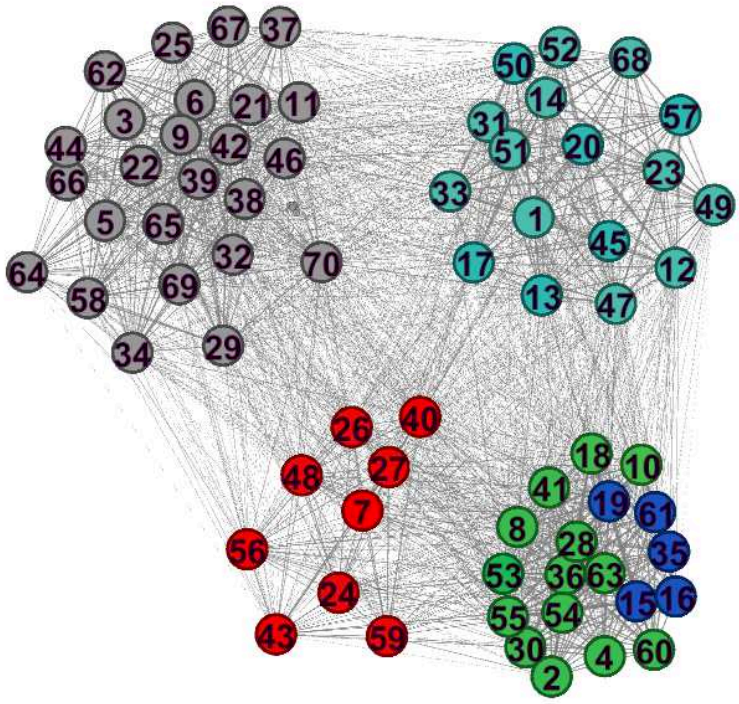

FIGURE 2.2 – Second simulated graph visualization with `Gephi`.

RMSE($\bar{\alpha}_1$)	RMSE($\bar{\alpha}_2$)	RMSE($\bar{\alpha}_3$)	RMSE($\bar{\alpha}_4$)	RMSE($\bar{\alpha}_5$)
0.036	0.026	0.013	0.057	0.001

TABLE 2.5 – Root Mean Squares Error of the parameter $\bar{\alpha}_q$ for the second simulated `SBM` with binomial output.

RMSE	$\bar{\pi}_{.1}$	$\bar{\pi}_{.2}$	$\bar{\pi}_{.3}$	$\bar{\pi}_{.4}$	$\bar{\pi}_{.5}$
$\bar{\pi}_{1.}$	0.02	0.05	0.08	0.01	0.09
$\bar{\pi}_{2.}$	0.05	0.03	0.08	0.08	0.002
$\bar{\pi}_{3.}$	0.08	0.08	0.05	0.04	0.04
$\bar{\pi}_{4.}$	0.01	0.08	0.04	0.01	0.05
$\bar{\pi}_{5.}$	0.09	0.002	0.04	0.05	0.04

TABLE 2.6 – Root Mean Squares Error of the parameter $\bar{\pi}_{qr}$ for the second simulated `SBM` with binomial output on edges.

All the obtained values in Table 2.6 are close to zero, wich means that the estimated parameters $\hat{\alpha}$ and $\hat{\pi}$ are close to the observed parameters $\bar{\alpha}$ and $\bar{\pi}$ respectively. That demonstrates the effectiveness of the proposed method. The time of convergence of the algorithm is 0.47 second (CPU Corel3 - 4GB RAM) which is satisfying.

Now, by calculating the average `ARI` between the simulated clustering results and the estimated clustering results obtained by using the proposed method, we obtain average `ARI`=0.83. This means a high agreement between the two partitions of the nodes.

2.6.2 Co-citation networks

Twitter network's data

The data consists of 154 tweets and 21 terms and has the form of a tweet-by-term matrix. The data is available online at http://www.rdatamining.com/data. For more explanation about this data, we refer the reader to (Zhao [2012]). We transform the tweet-by-term matrix into a term-by-term matrix based on the co-occurrence of term in the same tweets. The network associated to the matrix is an undirected network of 21 vertices and 130 edges, where each vertex is a term and there is an edge between a pair of terms if they co-occur together a least one time in the tweets. The graph associated with the network is visualized in Figure 2.3 using `Gephi` software with the layout algorithm Force Atlas.

We present in Table 2.7 the assortativity coefficient, the average clustering coefficient and the density of the twitter's network.

Assortativity	Average clustering coefficient	Density
0.24	0.78	0.62

TABLE 2.7 – Global characteristics of the twitter network's structure.

We can show in Table (2.7) that the assortativity coefficient is equal to 0.24 which is a positive value. That means that the terms in the tweets tends to occur with others that have equally high or equally low number of occurrence. The average clustering coefficient (transitivity) is equal to 0.78 which shows the completeness of the neighborhood of the vertices in the network. The density of the graph is equal to 0.62 which indicates that the graph of the network is dense. Note that the transitivity and the density value are close which means that the network is not highly clustered.

By applying our algorithm implemented in software `R`, we obtain that the network terms are grouped into three groups as shown in the following Table 2.8.

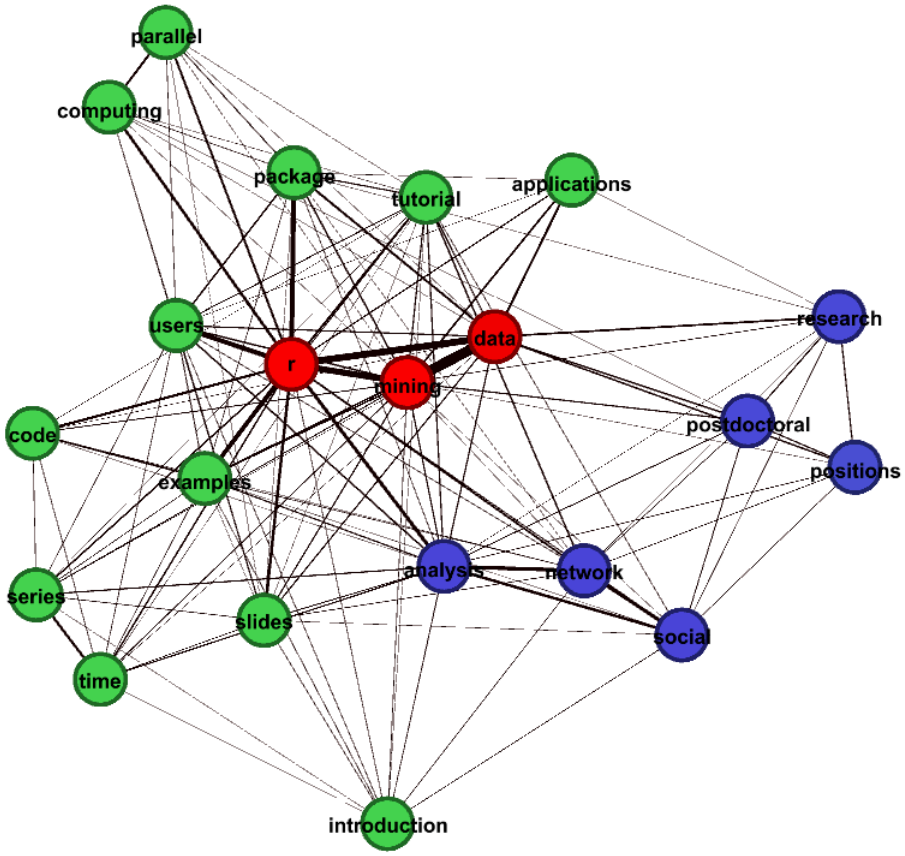

FIGURE 2.3 – Twitter network of terms visualization with `Gephi`.

Clusters	vertices
Red	r data mining
Blue	research postdoctoral positions analysis social network
Green	parallel computing time series code examples slides applications package users tutorial introduction

TABLE 2.8 – Grouping the terms of the twitter network into clusters.

We can show in Table 2.8 the classification of the twitter network's terms into clusters. Thus, the terms of each cluster are often cited together in the tweets.

Text mining through terms co-occurence network (Reuters-21578 dataset)

The Reuters-21578 dataset contains a collection of documents that appeared on Reuters newswire in 1987. The dataset is available online at http://kdd.ics.uci.edu/databases /reuters21578/reuters21578.html. For more explanation about this data, we refer the reader to (Lewis [1997]). We are interested in this example in 20 exemplary news articles from the Reuters-21578 dataset of topic crude. The data is available in the package `tm` (Feinerer et al. [2008]) of the software `R` under the name of `crude` data where all documents belong to the topic crude dealing with crude oil. We build a term-by-document matrix of the corpus `crude` by doing a text mining treatment. We interpret a term as important according to a simple counting of frequencies, we chose the frequent terms that co-occur at least six times in the documents. Then, we compute the correlations between them in the term-by-document matrix and we chose those out higher than 0.5. The figure visualizing the correlation between these terms is available in (Feinerer et al. [2008]).

We transform the term-by-document matrix into a one mode matrix which is the term-by-term matrix. The network associated to this matrix is an undirected network of 21 vertices and 97 edges, where each vertex is a term and there is an edge between a pair of terms if they co-occur together at least one time in the documents. The edge weights are represented in the obtained matrix where each cell indicates the number of documents where both the row and the column terms co-occur.

The graph associated with this network is visualized in Figure 2.4 using `Gephi` software with the layout algorithm Force Atlas. Table 2.9 presents some global characteristics of the structure of the associated graph with "unweighted" edges. We present the assortativity coefficient, the average clustering coefficient and the density of the network of terms of the Reuters-21578 corpus.

Assortativity	Average clustering coefficient	Density
0.23	0.84	0.51

TABLE 2.9 – Global characteristics of the structure of the network of terms of the Reuters-21578 corpus.

We can show in Table 2.9 that the assortativity coefficient is equal to 0.23 which is a positive value. That means that the terms presented in the documents of the reuters-21578 corpus tends to occur with other terms that have equally high or equally low number of occurrence. The average clustering coefficient (transitivity) is equal to 0.84 which shows the completeness of the neighborhood of the vertices in the network. The density of the graph is equal to 0.51 which indicates that the

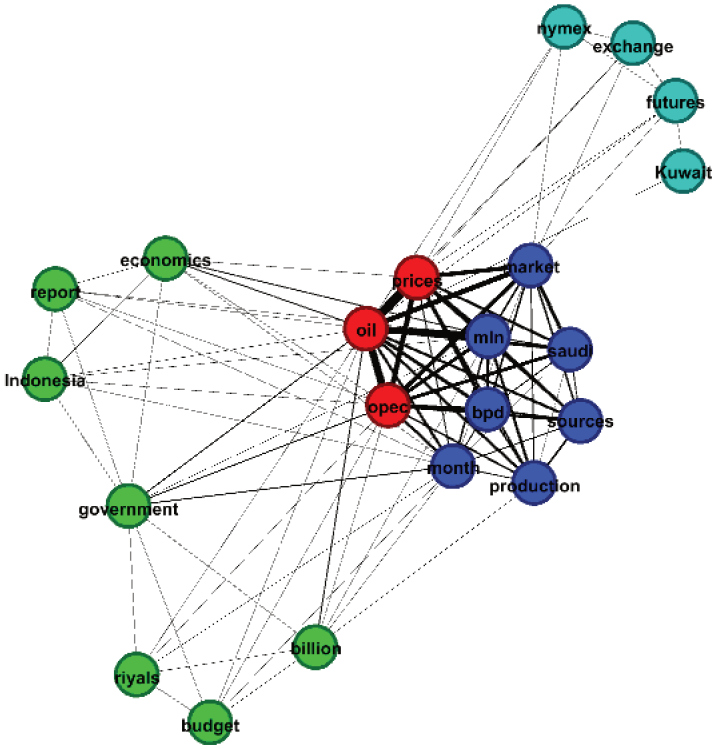

FIGURE 2.4 – Network of terms of the the Reuters-21578 corpus visualization with `Gephi`.

graph of the network is dense. Note that the transitivity and the density value are close which means that the graph is not highly clustered.

We apply our algorithm. We obtain that the terms are grouped into four clusters as presented in Table 2.10.

Clusters	Vertices
Red	oil opec prices
Blue	mln bpd month sources production saudi market
Green	billion budget riyals government economics Indonesia report
Cyan	exchange nymex futures Kuwait

TABLE 2.10 – Grouping the terms of the network of terms of the Reuters-21578 corpus into clusters using binomial `SBM`.

Table 2.10 shows the classification of the network's terms into clusters. Thus, the terms of each obtained clusters are frequently co-occurring together in the documents.

2.6.3 Social network : a benchmark dataset

Deep South network

The data was collected by Davies et al. [1941] in the Southern United State 1930s in order to report a comparative study of social in black and in white society. They are interested in the percentage of the contacts between individuals which have approximately the same class levels so they collect the deep South data which represents the participation of 18 white women in a series of 14 informal social events over a nine-month period. The data is available in the package `manet` in software `R` under the name `deepsouth` http://cran.r-projet.org/web/packages/manet/manet.pdf. For more explanation about this data, we refer the reader to (Linton [2003]). This data is considered as a benchmark in comparing social network analysis method. The authors focus on the analysis of two-mode data which means the women-by-event matrix data. The data is represented in Table 2.11.

Women	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14
Eyelin	1	1	1	1	1	1	0	1	1	0	0	0	0	0
Laura	1	1	1	0	1	1	1	1	0	0	0	0	0	0
Theresa	0	1	1	1	1	1	1	1	1	0	0	0	0	0
Brenda	1	0	1	1	1	1	1	1	0	0	0	0	0	0
Charlotte	0	0	1	1	1	0	1	0	0	0	0	0	0	0
Frances	0	0	1	0	1	1	0	1	0	0	0	0	0	0
Eleanor	0	0	0	0	1	1	1	1	0	0	0	0	0	0
Pearl	0	0	0	0	0	1	0	1	1	0	0	0	0	0
Ruth	0	0	0	0	0	0	1	1	1	0	0	1	0	0
verne	0	0	0	0	0	0	0	1	1	1	0	1	0	0
Myrna	0	0	0	0	0	0	0	1	1	1	0	1	0	0
katherine	0	0	0	0	0	0	0	1	1	1	0	1	1	1
Sylvia	0	0	0	0	0	0	1	1	1	1	0	1	1	1
Nora	0	0	0	0	0	1	1	0	1	1	1	1	1	1
Helen	0	0	0	0	0	0	1	1	0	1	1	1	0	0
Dorothy	0	0	0	0	0	0	0	1	1	0	0	0	0	0
Olivia	0	0	0	0	0	0	0	0	1	0	1	0	0	0
Flora	0	0	0	0	0	0	0	0	1	0	1	0	0	0

TABLE 2.11 – A two-mode representation of the deep South data.

The rows correspond to the Southern women and the columns are the events they attended. The value 1 in the Table indicates attendance of the woman at an event and the value 0 indicates not attending an event.

We transform the data into a single mode matrix which is the women-by-women matrix by multiplying the data matrix by its transpose. The network associated to this matrix is an undirected network of 18 vertices and 139 edges, where each vertex represents a Southern women among the 18 and there is an edge between a pair of women if they participate together in one of the 14 events a least. The

edge weights are represented in the obtained matrix where each cell indicates the number of events co-attended by both the row and the column women.

The graph associated with the obtained network is visualized in Figure 2.5 using Gephi software with the layout algorithm Force Atlas.

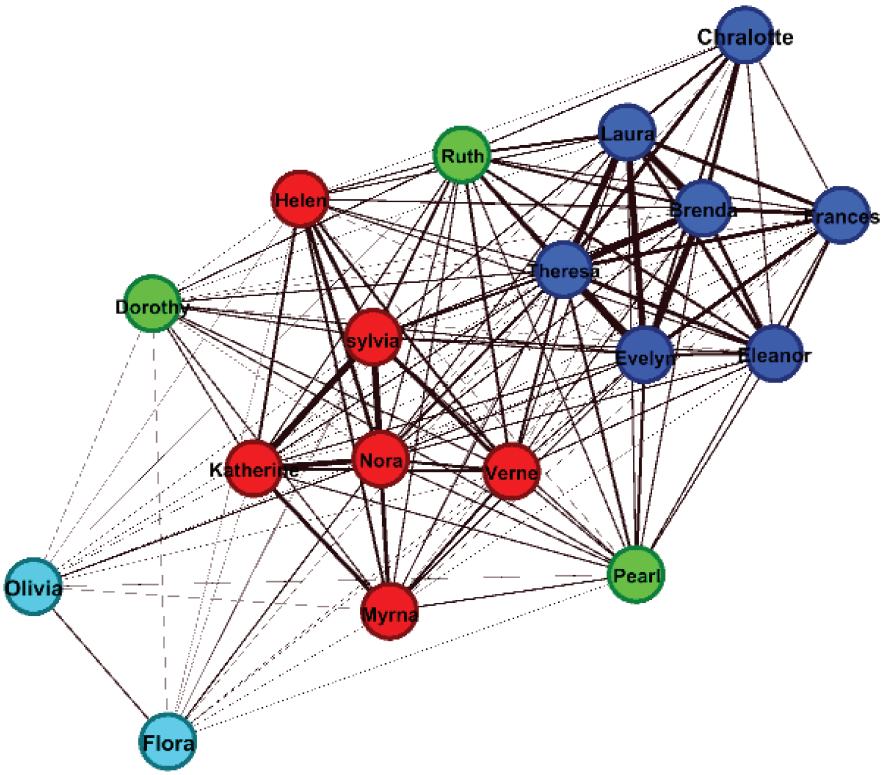

FIGURE 2.5 – Deep South network visualization with Gephi.

We present in Table 2.12 some global characteristics of the structure of the associated graph with "non weighted" edges. We present the assortativity coefficient, the average clustering coefficient and the density of the deep South network.

Assortativity	Average clustering coefficient	Density
0.23	0.84	0.51

TABLE 2.12 – Global characteristics of the deep South network's structure.

We can show in Table (2.12) that the assortativity coefficient is equal to 0.23

which is a positive value. That means that the Southern's women tends to participate to social events with other women that have equally high or equally low number of participation in the events. The average clustering coefficient (transitivity) is equal to 0.93 which shows the completeness of the neighborhood of the vertices in the network. The density of the graph is equal to 0.9 which indicates that the graph of the network is dense. Note that the transitivity and the density value are close which means that the graph is not highly clustered.

We apply our algorithm on the network to cluster the women into groups based on their occurrence in the events. The results are shown in Table 2.13.

Clusters	Vertices
Cyan	Olivia Flora
Blue	Evelyn Laura Theresa Brenda Charlotte Frances Eleanor
Red	Verne Myrna katherine Sylvia Nora Helen
Green	Pearl Ruth Dorothy

TABLE 2.13 – Grouping the women of the deep South network into clusters.

In table 2.13, each cluster represents the women which are frequently met together in the informal social events.

We compare in the following the results obtained by the proposed method to several already existing methods : `BGR74` proposed by Breiger [1974] and is based on algebraic approaches, `FRE92`, `FRE193` and `FR293` proposed by Freeman [1993] and Freeman [1994] and is based on various algorithms to search for an optimal partition and `OSB00` proposed by Osbourn and Martinez [1995] and is based on the algorithm `VERI`.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
BGR74	W	W	W	W	W	W	W		W	W	W	W	W	WW	WW		W	W
FRE92	W	W	W	W	W	W	W		W	W	W	W	W	W	W	W		
FR193	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
FR293	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
OSB00	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
bSBM	W	W	W	W	W	W	W	w	w	W	W	W	W	W	W	w	W	W

TABLE 2.14 – Clustering the women of the deep South network by different methods.

Table 2.14 shows the clusters obtained by different methods. At each line, The symbol "W" corresponds to women and all the W of the same color correspond to the women in the same cluster. The `bSBM` line corresponds to our method.

2.7 SBM with Poisson distributed weights

In this section, we define the `SBM` with Poisson distributed weights method in order to compare it later to the `SBM` with binomial distributed weights method.

2.7.1 Generation of the Poisson SBM data's

The network is assumed to be sampled as follows

— Each node i belongs to an unobserved group q among Q such as :

$$Z_{iq} \sim \mathbb{M}(1, \alpha = (\alpha_1, \ldots, \alpha_Q)).$$

— Each observed edge X_{ij} joining i and j is sampled from a Poisson distribution such as :

$$X_{ij}|Z_{iq}Z_{jl} = 1 \sim \mathbb{P}(\lambda_{qr}),$$

where $\lambda = (\lambda_{ql})_{ql}$ is the $Q \times Q$ block affinity matrix between the latent groups.

Thus, the Poisson stochastic blockmodel consists of the following parameters

— The latent variables $Z_i, \forall i \in \{1, \ldots, n\}$.

— The vector $\gamma = (\alpha, \lambda)$.

In the sequel, we are interested in estimating these parameters in a weighted undirected network. However, we affirm that all results obtained in this paper can be extended to directed networks, with or without self-loops.

2.7.2 Likelihood of the complete data

The likelihood of the complete data can be expressed as follows

$$\mathbb{P}_\gamma(X) = \mathbb{P}_\lambda(X|Z)\mathbb{P}_\alpha(Z),$$

where

$$\begin{aligned}
\mathbb{P}_\gamma(X|Z) &= \prod_{i<j}^{n}\prod_{q,l}^{Q} \mathbb{P}_{\lambda_{ql}}(X_{i,j}|Z_i, Z_j) \\
&= \prod_{i<j}^{n}\prod_{q,l}^{Q} \mathbb{P}_{\lambda_{ql}}(X_{i,j})^{Z_{iq}Z_{jl}} \\
&= \prod_{i<j}^{n}\prod_{q,l}^{Q} \left(\frac{e^{-\lambda_{ql}}\lambda_{ql}^{X_{ij}}}{X_{ij}!}\right)^{Z_{iq}Z_{jl}}.
\end{aligned}$$

According to (2.1),we have

$$\mathbb{P}_\alpha(Z) = \prod_{i}^{n}\prod_{q}^{Q} \alpha_q^{Z_{iq}}.$$

Therefore, the log-likelihood of the complete data can be expressed as :

$$\begin{aligned}\log \mathbb{P}_\gamma &= \log \mathbb{P}_\alpha(Z) + \log \mathbb{P}_\lambda(X|Z) \\ &= \sum_i \sum_q Z_{iq} \log \alpha_q + \sum_{i<j} \sum_{q,l} Z_{iq} Z_{jl} \log \mathbb{P}_{\lambda_{ql}}(X_{ij}) \\ &= \sum_i \sum_q Z_{iq} \log \alpha_q + \sum_{i<j} \sum_{q,l} Z_{iq} Z_{jl} (-\lambda_{ql} + X_{ij} \log \lambda_{ql}) - \log(X_{ij}!)). \quad (2.15)\end{aligned}$$

2.7.3 Variational EM inference

The log-likelihood of the given data X is intractable since it requires a summation over all possible value of Z as follows

$$\log \mathbb{P}_\gamma(X) = \log \sum_Z \mathbb{P}_\gamma(X, Z).$$

Thus, we propose to use an iterative algorithm to tackle this issue. Since the EM algorithm requires the computation of the probability of Z conditionally on X which is intractable because of the dependency of the edges in the networks, we propose to use the VEM algorithm. This algorithm overcomes the problem by maximizing a lower bound of the log-likelihood based on an approximation of the true conditional distribution of Z given X.

According to (2.15) and following the same steps used in the section 2.4.3, we can express the lower bound of the log-likelihood as follows

$$\begin{aligned}J_\gamma(R_X) \quad = \quad & \sum_i \sum_q \tau_{iq} \log \alpha_q + \sum_{i<j} \sum_{q,l} \tau_{iq} \tau_{jl} (-\lambda_{ql} + X_{ij} \log \lambda_{ql} - \log(X_{ij}!)) \\ & - \sum_i \sum_q \tau_{iq} \log \tau_{iq}. \quad (2.16)\end{aligned}$$

The algorithm VEM goes through two steps :

— Step 1 : we fix the parameter γ then, we maximize $J_\gamma(R_X)$ with respect to τ.

— Step 2 : we fix the parameter τ then, we maximize $J_\gamma(R_X)$ with respect to γ.

Estimation of the parameters

We are interested in the estimation of the parameters γ and τ. To estimate the parameter τ, we apply the E-step of the VEM algorithm as follows

E-step : By fixing the parameter γ and by maximizing the lower bound $J_\gamma(R_X)$ with respect to τ and under the condition $\sum_q \tau_{iq} = 1, \forall i \in \{1, \ldots, n\}$, we obtain

the estimation of τ by the following fixed point relation

$$\hat{\tau}_{iq} \propto \alpha_q \prod_j \prod_l \left(\frac{e^{-\lambda_{ql}} \lambda_{ql}^{X_{ij}}}{X_{ij}!} \right)^{\hat{\tau}_{jl}} . \tag{2.17}$$

The estimation of τ is obtained from (2.17) by iterating a fixed point algorithm until convergence.

Proof. The lower bound must be maximized with respect to τ under the constraint $\sum_q^Q \tau_{iq} = 1, \forall i \in \{1, \ldots, n\}$. As a consequence, using the Lagrange multiplier, we compute the derivative of $J_\gamma(R_X(Z)) + \lambda_i(\sum_q^Q \tau_{iq} - 1)$ with respect to τ_{iq}, for all $i \in \{1, \ldots, n\}$, $q \in \{1, \ldots, Q\}$ and λ_i. Recall that λ_i is the Lagrange multiplier.

According to (2.16), we have

$$\begin{aligned} J_\gamma(R_X(Z)) + \lambda_i(\sum_q^Q \tau_{iq} - 1) = & \sum_i \sum_q \tau_{iq} \log \alpha_q + \sum_{i<j} \sum_{q,l} \tau_{iq}\tau_{jl}(-\lambda_{ql} + X_{ij} \log \lambda_{ql} \\ & - \log(X_{ij}!)) - \sum_i \sum_q \tau_{iq} \log \tau_{iq} + \lambda_i(\sum_q \tau_{iq} - 1). \end{aligned} \tag{2.18}$$

By deriving (2.18) with respect to τ_{iq} and by taking this quantity equal to zero, we obtain :

$$\sum_l^Q \sum_{j=1, j\neq i}^n (-\lambda_{ql} + X_{ij} \log \lambda_{ql} - \log(X_{ij}!))\tau_{jl} + \log \alpha_q - \log \tau_{iq} - 1 + \lambda_i = 0.$$

Then, By deriving (2.18) with respect to λ_i and by taking this quantity equal to zero, we obtain :

$$\sum_q^Q \tau_{iq} - 1 = 0.$$

This leads to the following fixed point relation

$$\begin{aligned} \hat{\tau}_{iq} &= e^{-1+\lambda_i} \alpha_q \prod_j \prod_l \left(\frac{e^{-\lambda_{ql}} \lambda_{ql}^{X_{ij}}}{X_{ij}!} \right)^{\hat{\tau}_{jl}} \\ &\propto \alpha_q \prod_j \prod_l \left(\frac{e^{-\lambda_{ql}} \lambda_{ql}^{X_{ij}}}{X_{ij}!} \right)^{\hat{\tau}_{jl}} . \end{aligned} \tag{2.19}$$

Recall that $\propto$ means "proportional to" and $e^{(-1+\lambda_i)}$ is the normalizing constant. The equation (2.19) must be solved under the constraint $\sum_q^Q \tau_{iq} = 1$. The estimation of τ_{iq} is then obtained from (2.19) by iterating a fixed point algorithm until convergence. Note that the value of τ need to be normalized after each iteration :

$$\hat{\tau}_{iq} = \frac{\hat{\tau}_{iq}}{\sum_{l=1}^Q \hat{\tau}_{il}}.$$

□

The estimation of the parameters γ can be obtained through the M-step of the `VEM` algorithm as follows

M-step : By fixing the parameter τ and by maximizing the lower bound $J_\gamma(R_X)$ with respect to α and under the condition $\sum_q \alpha_q = 1$, we obtain the following estimation of α_q

$$\hat{\alpha}_q = \frac{1}{n}\sum_i \tau_{iq}.$$

The proof is given in the previous chapter.

Then, by maximizing the lower bound $J_\gamma(R_X)$ with respect to λ, we obtain the following estimation of λ_{ql}

$$\hat{\lambda}_{ql} = \frac{\sum_{i<j} \tau_{iq}\tau_{jl}X_{ij}}{\sum_{i<j} \tau_{iq}\tau_{jl}}.$$

Proof. we fix the parameters τ and α, then we maximize the lower bound (2.16) with respect to λ_{ql}. By deriving (2.16) with respect to λ_{ql} and by taking this quantity equal to zero, we obtain :

$$\sum_{i<j} \tau_{iq}\tau_{jl}\left(-1 + \frac{X_{ij}}{\lambda_{ql}}\right) = 0.$$

This leads to the following estimate of λ_{ql}

$$\hat{\lambda}_{ql} = \frac{\sum_{i<j} \tau_{iq}\tau_{jl}X_{ij}}{\sum_{i<j} \tau_{iq}\tau_{jl}}.$$

□

2.7.4 Model selection

We are interested here in determining the optimal number of clusters $\hat{Q}$ in the network. Since the number of clusters Q in the weighted network was fixed in the sections above, we develop in this section a criterion to select the optimal one.

As we have already seen in the section 2.5, Daudin et al. [2008] proposed the integrated classification likelihood (ICL) criterion to calculate the optimal number of clusters $\hat{Q}$ in the stochastic block model. The ICL is of the form :

$$\begin{aligned} ICL(Q) &= \sum_i \sum_q \hat{\tau}_{iq} \log \hat{\alpha}_q + \sum_{i<j} \sum_{q,l} \hat{\tau}_{iq}\hat{\tau}_{jl}(-\hat{\lambda}_{ql} + X_{ij} \log \hat{\lambda}_{ql} - \log(X_{ij}!)) - \frac{V_Q}{2} \log n \\ &= \sum_i \sum_q \hat{\tau}_{iq} \log \hat{\alpha}_q + \sum_{i<j} \sum_{q,l} \hat{\tau}_{iq}\hat{\tau}_{jl}(-\hat{\lambda}_{ql} + X_{ij} \log \hat{\lambda}_{ql} - \log(X_{ij}!)) \\ &\quad -\frac{1}{2}\left(\frac{Q(Q+1)}{2} \log \frac{n(n-1)}{2} + (Q-1)\log n\right), \end{aligned}$$

where V_Q is the total number of parameters of the model for the Q clusters.

The VEM algorithm is run for different values of Q. The optimal one is chosen such that the ICL is maximized.

2.7.5 Numerical comparison

This section aims to compare the results obtained by using the SBM with binomial distributed weights to those obtained by using the SBM with Poisson distributed weights. So, we resume the previous data examples present in section 2.6, we apply the Poisson SBM to show numerically the obtained results and then we compare them to the binomial SBM.

Simulated data

We resume here the first simulated data example by taking the parameter $\bar{\lambda}$ equal to $m\bar{\pi}$. This data is detailed in the section 2.6 and the associated graph is given in Figure 2.1. By applying the Poisson SBM algorithm implemented in R programming language, we obtain that the vertices of the network are grouped into three clusters as shown in Table 2.15.

Clusters	Vertices
Red	3 8 9 14 20
Blue	4 5 6 7 10 11 12 13 15 17
Green	1 2 16 18 19

TABLE 2.15 – Grouping first simulated graph vertices into clusters using Poisson SBM

Table 2.15 shows clearly that the nodes of the first simulated graph are split into three clusters which are the same as the three clusters shown in Table 2.1. Therefore, the binomial SBM and the Poisson SBM provide the same clustering of the vertices of the simulated data.

We sample now $S = 100$ random graphs according to mixture model. Then, we calculate in table 2.16 and 2.17 the RMSE of the parameters $\bar{\alpha}_q$ and $\bar{\lambda}_{ql}$ respectively.

Table 2.16 shows the values of the RMSE of the parameters $\bar{\alpha}_q$ for $q \in \{1, 2, 3\}$. By comparing them to those obtained by applying the binomial SBM in Table 2.2, we can show that the values obtained by the binomial SBM are closer to zero. Thus, the estimated parameter $\hat{\alpha}$ obtained by the binomial SBM is closer to the observed parameter $\bar{\alpha}$ than the $\hat{\alpha}$ obtained by the Poisson SBM.

RMSE($\bar{\alpha}_1$)	RMSE($\bar{\alpha}_2$)	RMSE($\bar{\alpha}_3$)
0.12	0.03	0.033

TABLE 2.16 – Root Mean Squares Error of the parameter $\bar{\alpha}_q$ for the first simulated data using the Poisson SBM model.

RMSE	$\bar{\lambda}_{.1}$	$\bar{\lambda}_{.2}$	$\bar{\lambda}_{.3}$
$\bar{\lambda}_{1.}$	0.06	0.09	0.12
$\bar{\lambda}_{2.}$	0.09	0.1	0.16
$\bar{\lambda}_{3.}$	0.12	0.16	0.15

TABLE 2.17 – Root Mean Squares Error of the parameter $\bar{\pi}_{qr}$ for the first simulated data using the Poisson SBM model.

Table 2.17 shows the values of the RMSE of the parameters $\bar{\lambda}_q$ for $\{q,l\} \in \{1,2,3\}\times\{1,2,3\}$. By comparing them to those obtained by applying the binomial SBM in Table 2.3, we can show that the values obtained by the binomial SBM are closer to zero. Thus, the estimated parameter $\hat{\pi}$ obtained by the binomial SBM is closer to the observed parameter $\bar{\pi}$ than $\hat{\lambda}$ obtained by the Poisson SBM is close to the observed parameter $\bar{\lambda}$.

We apply now the Poisson SBM implemented in R programming language on the second simulated data example by taking $\bar{\lambda}$ equal to $m\bar{\pi}$. The data is detailed in section 2.6 and the associated graph is given in 2.2. Results shows that the vertices are grouped into five clusters as shown in Table 2.18.

Clusters	Vertices
Cyan	1 12 13 14 17 20 23 31 33 45 47 49 50 51 52 57 68
Gray	3 5 6 9 11 21 22 25 29 32 34 37 38 39 42 44 46 58 62 64 65 66 67 69 70
Green	2 4 8 10 18 28 30 36 41 53 54 55 60 63
Blue	15 16 19 35 61
Red	7 24 26 27 40 43 48 56 59

TABLE 2.18 – Grouping second simulated graph vertices into clusters using Poisson SBM.

Table 2.18 shows clearly that the nodes of the second simulated graph are split into five clusters which are the same as the five clusters shown in Table 2.4. Therefore, the binomial SBM and the Poisson SBM provide the same clustering of the vertices of the network.

Twitter network's data

The Twitter network's data is detailed in the section 2.6.2 and is available in http://www.rdatamining.com/data. The associated graph is given in 3.2. By applying the Poisson SBM algorithm implemented in R programming language, we obtain that the terms are grouped into two clusters as presented in the table 2.19.

Clusters
r research postdoctoral positions analysis network
parallel computing time series code examples slides applications package users tutorial introduction data mining
social

TABLE 2.19 – Grouping the terms of the twitter network into clusters using Poisson SBM.

We can show in Table 2.19 the distribution of the twitter network's terms into clusters which means that the terms of each cluster are often cited together in the Tweets.

We define the total variation distance between two probability distributions μ and ν on the numerable sample space Ω by

$$d_{TV}(\mu, \nu) = \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)|. \tag{2.20}$$

The mean (over the whole set of edges) of the total variation distance (2.20) between the binomial and the Poisson distribution is equal to $md_{TV} = 4.5$ which means that the two approaches are not close and then the two fitted model are so different.

Reuters-21578 data

The data is detailed in the section 2.6.2 and is available in http://kdd.ics.uci.edu/databases/reuters21578/reuters 21578.html. The associated graph is given in 2.4. By applying the Poisson SBM algorithm implemented in R programming language, we obtain that the terms are grouped into two clusters as presented in the table 2.20.

Table 2.20 shows the distribution of the network's terms into clusters which means that the terms of each cluster are often cited together in the documents.

The mean of the total variation distance (2.20) between the binomial and the Poisson distribution is equal to $md_{TV} = 6.5$ which means that the two models are different.

Clusters
oil opec prices mln bpd sources production saudi market Kuwait
billion budget exchange futures riyals government economics indonesia month nymex report

TABLE 2.20 – Grouping the terms of the network of terms of the Reuters-21578 corpus into clusters using Poisson SBM.

Deep South network

The data is detailed in the section 2.6.3 and is available in the package manet in software R under the name deepsouth http://cran.r-projet.org/web/packages/manet/manet.pdf. The associated graph is given in 2.5. By applying the Poisson SBM algorithm implemented in R programming language, we obtain that the terms are grouped into three clusters as presented in the table 2.21.

Clusters
Olivia Flora
Evelyn Laura Theresa Brenda Charlotte Frances Eleanor Pearl Ruth
Verne Myrna katherine Sylvia Nora Helen Dorothy

TABLE 2.21 – Grouping the women of deep South network into clusters using Poisson SBM.

Table 2.21 shows the three clusters obtained by applying the Poisson SBM. Each cluster represents the women which are frequently meeting together in the informal social events.

We present in the following a comparative table 2.22 between the Poisson SBM and different methods detailed in section 2.6.3. Note that bSBM means the

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
BGR74	W	W	W	W	W	W	W		W	W	W	W	W	WW	WW		W	W
FRE92	W	W	W	W	W	W	W		W	W	W	W	W	W	W	W		
FR193	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
FR293	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
OSB00	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
bSBM	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
PSBM	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

TABLE 2.22 – Clustering of the women of the deep South network by different methods.

binomial SBM while PSBM means the Poisson SBM. Table 2.22 shows the clusters obtained by using different methods. Recall that at each line, the symbol "W" corresponds to women and all the W of the same color correspond to the women in the same cluster.

The mean of the total variation distance (2.20) between the binomial and the Poisson distribution is equal to $md_{TV} = 3.4$ which means that the two models are different.

Introduction en Français du Chapitre 3

Ceci est une introduction en français du chapitre "Variational Bayesian Inference in Binomial Stochastic Block model for Weighted Networks".

Introduction

Les modèles à blocs stochastiques sont des modèles statistiques largement utilisés dans l'analyse de réseaux sociaux. Ils visent à regrouper les noeuds du réseau étudié dans des clusters. Le modèle `SBM` indique que chaque noeud du réseau observé appartient à une certaine classe appelée aussi cluster et que la probabilité pour que deux noeuds soient connectés dépend de la classe à laquelle ils appartiennent.

Ces dernières années, des efforts ont été faits pour développer ces modèles afin de traiter les réseaux pondérés. Dans ces réseaux, les liens entre les noeuds sont affectées par des poids qui représentent les forces ou l'intensité de ces liens.

Dans le chapitre précédent, nous avons développé un modèle `SBM` binomial pour traiter les réseaux ayant des arêtes pondérées. On a développé un algorithme espérance maximisation variationnel (`VEM`) pour estimer les paramètres du modèle et les variables latentes ainsi que pour regrouper les noeuds du réseau dans des clusters homogènes.

Dans ce chapitre, nous proposons un processus d'inférence entièrement bayésien, basé sur des priors informatifs plausibles. Ce processus est indépendant des autres algorithmes de prétraitement des valeurs de départ pour l'affectation des noeuds à des clusters et a pour but d'estimer les paramètres du modèle `SBM` binomial ainsi que de regrouper les sommets du réseau dans des clusters. Notre méthode estime la vraisemblance marginale des modèles probabilistes avec des variables latentes. Elle construit et optimise une borne inférieure sur la vraisemblance marginale en utilisant le calcul variationnel, ce qui donne un algorithme itératif qui généralise l'algorithme espérance maximisation (`EM`) en gardant les distributions *à posteriori* sur les variables latentes et les paramètres. Cet algorithme est appelé

l'algorithme espérance maximisation variationnel Bayésien (VBEM). Il calcule (approximativement) la totalité de la distribution *à posteriori* des paramètres et des variables latentes. De plus, il possède la même structure alternée que l'algorithme EM, basé sur un ensemble d'équations emboîtées (mutuellement dépendantes) qui ne peuvent pas être résolues analytiquement.

Motivation

Les méthodes variationnelles connaissent du succès dû à leur facilité d'utilisation et à leur rapidité d'exécution dans des cas d'inférence difficile à traiter avec les méthodes classiques (méthodes de Monte Carlo par Chaîne de Markov (MCMC) par exemple). Or, dans les estimations bayésiennes, les lois *a posteriori* ne sont pas toujours accessibles. De même pour les méthodes de Monte-Carlo par Chaîne de Markov. Les méthodes variationnelles bayésiennes permettent de calculer directement (et rapidement) une approximation des lois *a posteriori*. Elles constituent une famille de techniques permettant d'approximer les intégrales intraitables résultantes de l'inférence bayésienne. Elles sont généralement utilisés dans des modèles statistiques complexes constitués de variables observées (appelées aussi données), ainsi que de paramètres inconnus (non observés) et de variables latentes (non observées), avec des équations modélisant les relations entre ces variables aléatoires.

Nous développons dans ce chapitre, un modèle SBM binomial pour traiter le cas d'un réseau pondéré, où chaque arête, joignant une paire de noeuds, est affectée d'une valeur représentant la force du lien entre cette paire de noeuds. Cette question est motivée par l'étude des réseaux de co-citations dans un contexte de fouille de texte. Dans ce type de réseaux, le poids associé à une arête joignant deux termes correspond au nombre de documents inclus dans le corpus citant simultanément ces deux termes. Nous avons introduit dans le chapitre précédent un exemple d'un réseau de co-citation (voir figure A.1). Ce réseau est constitué d'articles qui sont les noeuds de ce réseau et de liens entre chaque paire d'articles. Ces liens représentent les arêtes du réseau. Ils sont pondérés en fonction du nombre de co-citation de ces deux articles ensemble.

Puis, nous utilisons une méthode variationnelle bayésienne pour estimer les paramètres du modèle SBM binomial ainsi que pour regrouper les noeuds dans des clusters homogènes. Cette méthode permet de fournir une approximation de la distribution *a posteriori* des variables non observées, soit les paramètres inconnus ou les variables latentes, afin de réaliser une inférence statistique sur ces variables. Elle permet de déterminer une borne inférieure de la vraisemblance marginale des variables observées. Cette borne inférieure est utilisée pour trouver le nombre optimal de clusters correspondant au modèle en effectuant une sélection du modèle.

A la fin, nous introduisons un corpus d'entretiens avec des mineurs migrants, de la région subsaharienne à la côte européenne méditerranéenne. Ce corpus contient

les témoignages d'une centaine de mineurs en migration ayant acceptés de répondre à un entretien semi-dirigé. Ces témoignages en français ont été mis en textes numériques. Nous choisissons ensuite les 25 termes les plus pertinents pour le spécialiser à partir d'une liste de termes classés par ordre de fréquence dans l'ensemble de corpus. Nous considérons une matrice termes-documents ayant les 25 termes choisis en ligne et les entretiens en colonne. Ensuite, nous convertissons cette matrice en une matrice terme-terme. Dans cette matrice, la valeur associée à chaque paire de termes représente le nombre d'entretiens utilisant ces deux termes. Le réseau associé à cette matrice est un réseau pondéré dont les noeuds sont les termes et les arêtes joignant chaque paire de termes sont pondérées en fonction du nombre d'entretiens utilisant ces deux termes. Nous classifions à la fin ces termes en utilisant notre approche puis nous comparons les résultats obtenus par cette méthode avec les résultats obtenus par la méthode `VEM`.

Structure du Chapitre

Dans ce chapitre, nous définissons un réseau non orienté pondéré et nous introduisons quelques notations dans la section 3.2. Nous introduisons le modèle à blocs stochastiques avec des arêtes pondérées distribuées selon une loi binomiale. Dans la section 3.3, nous introduisons les distributions des priors conjugués non informatives pour les paramètres du modèle. Dans la section 3.4, nous mettons en oeuvre une inférence dans le modèle à blocs stochastiques binomial dans le cadre variationnel Bayésien. Ensuite, l'algorithme variationnel bayésien est présenté à la sous section 3.4.1. Dans la section 3.5, nous adoptons un critère de sélection pour trouver le nombre optimal de clusters correspondant au modèle proposé, puis dans la section 3.6, nous reprenons les mêmes données introduites dans le chapitre précédent et nous réalisons des applications de notre modèle afin de comparer les deux méthodes. Enfin, nous introduisons une application de la méthode proposée à l'aide des données réelles dans la section 3.7 puis nous comparons les résultats obtenus en utilisant la méthode proposée avec les résultats obtenus en utilisant la méthode utilisée dans le chapitre précédent.

Chapitre 3

Variational Bayesian Inference in Binomial Stochastic Block model for Weighted Networks

3.1 Introduction

Stochastic block models are statistical models widely used in the analysis of social networks. They aim at grouping the nodes of the network in clusters. The SBM model indicates that each node of the observed network belongs to a certain class called cluster and that the probability that two nodes are connected depends on the class to which they belong.

Recently, efforts have been made to extend these models to the weighted networks. In these networks, links between nodes are affected by weights that represent their strength or intensity.

In the previous chapter, we have developed a binomial SBM model to handle networks with edges weights. We have developed a variational expectation maximization (VEM) algorithm to estimate the parameters of the model and the latent variables as well as to classify the nodes of the network in homogeneous clusters.

In this chapter, we propose a Bayesian inference approach, based on plausible informative priors. This approach is independent of the other preprocessing algorithms of starting values for node assignment to clusters. Furthermore, it aims at estimating the parameters of the binomial SBM model as well as to classify the vertices of the network. This method estimates the marginal likelihood of the probabilistic models with latent variables. It builds and optimizes a lower bound of the marginal likelihood using variational calculus, which gives an iterative algorithm that generalize the expectation maximization (EM) algorithm by keeping the posterior distributions on the latent variables and the parameters. This algorithm

is called variational Bayesian expectation maximization (VBEM) algorithm. It calculates (approximately) the totality of the posterior distribution of the parameters and the latent variables. Moreover, it has the same alternating structure as the EM algorithm, based on a set of embroidered equations (mutually dependents) that can not be solved analytically.

3.1.1 Motivation

Variational methods have achieved a success due to their ease of use and their speed of execution in the cases of inference that may be difficult to deal with classical methods (for example, Markov Chain Monte Carlo methods (MCMC). However, in Bayesian estimates, posterior distributions are not always accessible. The same for Monte Carlo methods by Markov's Chain. The variational Bayesian methods allow us to compute directly (and rapidly) an approximation of the posterior distributions. They constitute a family of techniques allowing to approximate the intractable integrals resulting from the Bayesian inference. They are generally used in complex statistical models consisting of observed variables (also called given variables), as well as of unknown parameters (unobserved parameters) and latent variables (unobserved), with equations modeling the relationships between these random variables.

We develop in this chapter a binomial SBM model for weighted network, where each edge in the network, joining a pair of nodes, is assigned to an integer value representing the strength of the link between this pair of nodes. This question is motivated by the study of co-citation networks in a context of text mining. In this type of network, the weight associated to an edge joining two terms corresponds to the number of documents included in the corpus simultaneously citing these two terms. We have introduced in the previous chapter an example of a co-citation network (see figure A.1). This network consists of papers that are the nodes of this network and links between each pair of papers. These links represent the edges of the network. They are weighted according to the co-citation number of these two terms together.

Then, we use a variational Bayesian method to estimate the parameters in a binomial SBM models and to classify the nodes into homogeneous clusters. This method provides an analytical approximation of the posterior distribution of the unobserved variables, either unknown parameters or latent variables, to obtain a statistical inference on these variables. Furthermore, It allows to determine a lower bound of the marginal likelihood of the observed variables. This lower bound is used to find the optimal number of clusters corresponding to the model by performing a selection of the model.

At the end, we introduce a corpus of interviews with migrant minors, from Sub-Sahara to the European Mediterranean coast. This corpus contains the testimonies

of a hundred migrant minors who have accepted to answer to a semi-directed interview. These testimonials have been put into digital texts. Then, we choose the most relevant terms from a list of terms ranked in order of frequency in the corpus. We consider a term-document matrix where rows are terms and columns are interviews. Then, we convert this matrix into a term-term matrix. In this matrix, the value associated with each pair of terms represents the number of interviews using these two terms. The network associated with this matrix is a weighted network whose nodes are the terms and the edges joining each pair of terms are weighted by the number of interviews using these two terms. At the end, we classify these terms using our approach then we compare the results obtained by using this approach to those obtained by using the `VEM`.

3.2 Definition of the Model

A weighted undirected network is defined by its set of N nodes $[N] = \{1, ..., N\}$ for all $N \geq 1$ and by its edge-weighted symmetric matrix X defined as follows

$$\begin{cases} X_{ij} = m_{ij} & \text{if } i \text{ and } j \text{ interact with an interaction strength } m_{ij} \\ X_{ij} = 0 & \text{otherwise.} \end{cases}$$

We choose a fixed number of blocks in the graph equal to Q.

The network is assumed to be generated as follows

— Each node i in the network is associated with a binary latent vector Z_i sampled from a multinomial distribution such as :

$$Z_i \sim \mathcal{M}(1, \alpha = (\alpha_1, \ldots, \alpha_Q)),$$

where $\mathcal{M}$ is the multinomial distribution and α is the vector of class proportion such as $\sum_q \alpha_q = 1$. Moreover, since a node i can belong to a single cluster, then all Z_i, for $i \in \{1, \ldots, N\}$, are i.i.d. and therefore $\sum_{q=1}^{Q} Z_{iq} = 1$. Furthermore, we have $\forall \{i, q\} \in \{1, \ldots, N\} \times \{1, \ldots, q\}$,

$$Z_{iq} = \begin{cases} 1 & \text{if node } i \text{ belongs to cluster } q \\ 0 & \text{otherwise.} \end{cases}$$

The matrix Z is composed of Z_{iq} and is of dimension $N \times Q$.

— Each observed edge X_{ij} joining node i, that belongs to group q, to node j, that belongs to group l, is sampled from a binomial distribution such as :

$$X_{ij} | Z_{iq} Z_{jl} = 1 \sim \mathcal{B}(m, \pi_{ql}),$$

where the parameter m indicates the maximal weight associated to the edges of the network and $(\pi_{ql})_{ql}$ is the connection probability between the clusters q and l. The matrix $\pi = (\pi_{ql})_{ql}$ represents the $Q \times Q$ matrix of connection probabilities between all the latent groups.

In the sequel, we assume that the nodes are not connected to themselves which means that there is no edges joining the node to itself so that for all $i \in \{1, \ldots, N\}$, we have $X_{ii} = 0$.

In the following, we treat the case of weighted undirected networks. However, we affirm that the obtained results can be extended to directed networks, with or without self-loop.

3.3 Variational Bayesian Approach

To realize an inference with variational Bayesian expectation maximization (`VBEM`) methods, a Bayesian view of the (`SBM`) is retained . The idea of the Bayesian treatment of the `SBM` is to set prior distributions for the unknown parameters of the `SBM`. In this case, the parameters of the model are treated as random variables.

In this section, we adopt the Bayesian approach developed by Nowicki and Snijders [2001]. We put prior distributions on the parameters α and π of the stochastic blockmodel We rely on Latouche et al. [2012] to specify some non informative conjugate priors for the model parameters. To simplify the computations, we use conjugate priors to facilitate the computation. Since Z_i is sampled from a multinomial distribution, we choose a Dirichlet distribution to model the mixing coefficient α such as :

$$\begin{aligned} \mathbb{P}(\alpha|n^0 = (n_1^0, \ldots, n_Q^0)) &= \mathrm{Dir}(n^0 = (n_1^0, \ldots, n_Q^0)) \\ &= \frac{\Gamma(\sum_{q=1}^{Q} n_q^0)}{\Gamma(n_1^0) \ldots \Gamma(n_Q^0)} \prod_{q=1}^{Q} \alpha_q^{n_q^0 - 1}, \end{aligned}$$

where the prior number of vertices in the q-th component of the mixture n_q^0 is regularly defined in the literature as $n_q^0 = \frac{1}{2}$ for all $q \in \{1, \ldots, Q\}$. Thus, the Dirichlet distribution corresponds to the non informative distribution of Jeffreys [1946].

Based on Latouche et al. [2012], since $X_{ij}|Z_{iq}Z_{jl} = 1$ is sampled from a binomial distribution, we use independent Beta priors to model the connectivity matrix π

such as :

$$\begin{aligned}\mathbb{P}(\pi|\beta^0=(\beta^0_{ql})_{ql},\gamma^0=(\gamma^0_{ql})_{ql}) &= \prod_{q\leq l}^{Q}\text{Beta}(\beta^0_{ql},\gamma^0_{ql})\\ &= \prod_{q\leq l}^{Q}\frac{\Gamma(\beta^0_{ql}+\gamma^0_{ql})}{\Gamma(\beta^0_{ql})\Gamma(\gamma^0_{ql})}\pi_{ql}^{\beta^0_{ql}-1}(1-\pi_{ql})^{\gamma^0_{ql}-1},\end{aligned}$$

where β^0_{ql} is the prior number of edges joining the vertices of cluster q and l while γ^0_{ql} is the prior number of non-edges joining the vertices of cluster q and l. These parameters are regularly defined in the literature as $\beta^0_{ql}=\frac{1}{2}$ and $\gamma^0_{ql}=\frac{1}{2}$, for all $q,l\in\{1,\ldots,Q\}$. Thus, The product of Beta distribution corresponds to a product of non informative distribution of Jeffreys [1946].

Since we consider here the case of undirected networks, the connection probability matrix π is symmetric. Thus, the terms of the upper triangular matrix are identical to those of the lower triangular matrix. For that, we compute over $q\leq l$ instead of the product over q,l. Note that in the case of directed graph, this product over $q\leq l$ must be replaced by the product over q,l.

We are interested in estimating the following the parameters in a Bayesian binomial stochastic block model :

— The latent variables $Z_i,\forall i\in\{1,\ldots,N\}$.
— The vector $n^0=(n^0_1,\ldots,n^0_Q)$.
— The two matrix $\beta^0=(\beta^0_{ql})_{ql}$ and $\gamma^0=(\gamma^0_{ql})_{ql}$.

3.4 Estimation in Bayesian SBM

In this section, we describe the proposed method to estimate the parameters of the binomial SBM model.

The dataset here is incomplete since there are some latent variables that influence the distribution of the data and the formation of the clusters within the network. The log-likelihood of the incomplete data can not be factorized and has a prohibitive calculation cost since it requires the integration over all the possible values of the latent variable Z. Furthermore, because of the dependency structure on the observed edges of the graph, the distribution $\mathbb{P}(Z|X,\alpha,\pi)$ can not be factorized. Thus, the EM algorithm is intractable here since it requires the computation of $\mathbb{P}(Z|X,\alpha,\pi)$.

We are interested here in the approximation of the full distribution $\mathbb{P}(Z,\alpha,\pi|X)$. Following Attias [1992] and Svensén and Bishop [2004], we rely on a variational decomposition of the integrated observed-data log-likelihood as follows

$$\log\mathbb{P}(X)=J(q(Z,\alpha,\pi))+\text{KL}(q(Z,\alpha,\pi)\parallel\mathbb{P}(Z,\alpha,\pi|X)),\tag{3.1}$$

where

$$J(q(Z,\alpha,\pi)) = \sum_{Z} \int\int q(Z,\alpha,\pi) \log \frac{\mathbb{P}(X,Z,\alpha,\pi)}{q(Z,\alpha,\pi)}\, d\alpha\, d\pi \tag{3.2}$$

and

$$\mathrm{KL}(q(Z,\alpha,\pi) \parallel \mathbb{P}(Z,\alpha,\pi|X)) = -\sum_{Z} \int\int q(Z,\alpha,\pi) \log \frac{\mathbb{P}(Z,\alpha,\pi|X)}{q(Z,\alpha,\pi)}\, d\alpha\, d\pi, \tag{3.3}$$

is the Kullback-Leibler divergence between $\mathbb{P}(Z,\alpha,\pi|X)$ which is the true conditional distribution of Z given X and $q(Z,\alpha,\pi)$ which is an approximate distribution of $\mathbb{P}(Z,\alpha,\pi|X)$. It measures the closeness of these two distributions. Furthermore, the Kullback-Leibler divergence is a non-negative measure :

$$\mathrm{KL}(q(Z,\alpha,\pi) \parallel \mathbb{P}(Z,\alpha,\pi|X)) \geq 0. \tag{3.4}$$

By combining the two equations (3.1) and (3.4), we obtain

$$\log \mathbb{P}(X) \geq J(q(Z,\alpha,\pi)).$$

Therefore, $J(q(Z,\alpha,\pi))$ is a lower bound of $\log \mathbb{P}(X)$.

Since $\log \mathbb{P}(X)$ does not depend on $q(Z,\alpha,\pi)$, minimizing equation (3.3) is equivalent to maximizing the lower bound equation (3.2).

To obtain a tractable algorithm, we assume that $q(Z,\alpha,\pi)$ can be factorized over α, π and the latent variable Z as follows

$$\begin{aligned} q(Z,\alpha,\pi) &= q(\alpha)q(\pi)\prod_{i=1}^{N} q(Z_i) \\ &= q(\alpha)q(\pi)\prod_{i=1}^{N} h(Z_i;\tau_i). \end{aligned} \tag{3.5}$$

where $\{\tau_i \in [0,1]^Q, i = 1,\ldots,N\}$ are the variational parameters associated with $\{Z_i, i = 1,\ldots,N\}$ such as $\sum_q \tau_{iq} = 1,\ \forall i \in \{1,\ldots,N\}$ and h is the multinomial distribution.

The variational Bayesian expectation maximization algorithm rely on two steps :

— Variational Bayesian E-step : we fix $q(\alpha)$ and $q(\pi)$ then we calculate $q(Z_i)$ by maximizing the lower bound (3.2).

— Variational Bayesian M-step : We calculate the approximations of the distributions $q(\alpha)$ and $q(\pi)$ by fixing $q(Z_i)$ and then by maximizing the lower bound (3.2) with respect to $q(\alpha)$ and $q(\pi)$ respectively.

We are interested first in determining the approximation of the distributions $q(\alpha)$ and $q(\pi)$.

By maximizing the lower bound (3.2) with respect to $q(\alpha)$, we obtain that the approximation of the distribution $q(\alpha)$ is a Dirichlet distribution as follows

$$q(\alpha) = \texttt{Dir}(n), \tag{3.6}$$

where

$$n_q = n_q^0 + \sum_{i=1}^{N} \tau_{iq}, \ \forall q \in \{1, \dots, Q\}. \tag{3.7}$$

Proof. According to (3.2), we have

$$\begin{aligned}
J(q(Z,\alpha,\pi)) &= \sum_Z \int\int q(Z,\alpha,\pi) \log \frac{\mathbb{P}(X,Z,\alpha,\pi)}{q(Z,\alpha,\pi)} \, d\alpha \, d\pi \\
&= \sum_Z \int\int (q(Z,\alpha,\pi) \log \mathbb{P}(X,Z,\alpha,\pi) - q(Z,\alpha,\pi) \log q(Z,\alpha,\pi)) \, d\alpha \, d\pi \\
&= \mathbb{E}_{Z,\alpha,\pi}(\log \mathbb{P}(X,Z,\alpha,\pi)) - \mathbb{E}_{Z,\alpha,\pi}(\log q(Z,\alpha,\pi)) \\
&= \mathbb{E}_{Z,\pi}(\log \mathbb{P}(X|Z,\pi)) + \mathbb{E}_{Z,\alpha}(\log \mathbb{P}(Z|\alpha)) + \mathbb{E}_{\alpha}(\log \mathbb{P}(\alpha)) \\
&\quad + \mathbb{E}_{\pi}(\log \mathbb{P}(\pi)) - \sum_{i=1}^{N} \mathbb{E}_{Z_i}(\log q(Z_i)) - \mathbb{E}_{\alpha}(\log q(\alpha)) - \mathbb{E}_{\pi}(\log q(\pi)).
\end{aligned} \tag{3.8}$$

By deriving (3.8) with respect to $q(\alpha)$, and by taking this quantity equal to zero, we obtain :

$$\begin{aligned}
\log q(\alpha) &= \mathbb{E}_Z(\log \mathbb{P}(Z|\alpha)) + \log \mathbb{P}(\alpha) + \text{cst} \\
&= \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \log \alpha_q + \sum_{q=1}^{Q} (n_q^0 - 1) \log \alpha_q + \text{cst} \\
&= \sum_{q=1}^{Q} \left(n_q^0 - 1 + \sum_{i=1}^{N} \tau_{iq} \right) \log \alpha_q + \text{cst}.
\end{aligned} \tag{3.9}$$

By taking the exponential of (3.9), we obtain

$$\begin{aligned}
q(\alpha) &= e^{\left(\sum_{q=1}^{Q} \left(n_q^0 - 1 + \sum_{i=1}^{N} \tau_{iq} \right) \log \alpha_q + \text{cst} \right)} \\
&= \text{cst} \prod_{q=1}^{Q} \alpha_q^{\left(n_q^0 - 1 + \sum_{i=1}^{N} \tau_{iq} \right)}.
\end{aligned}$$

Thus, we obtain the Dirichlet distribution (3.6). □

However, by maximizing the lower bound (3.2) with respect to $q(\pi)$, we obtain that the approximation of the distribution $q(\pi)$ is a product of Beta distribution as follows

$$q(\pi) = \prod_{q \leq l}^{Q} \texttt{Beta}(\beta_{ql}, \gamma_{ql}), \tag{3.10}$$

where

$$\beta_{ql} = \beta^0_{ql} + \sum_{i\neq j}^{N} \tau_{iq}\tau_{jl}X_{ij},\ \forall q \neq l \in \{1,\dots,Q\}, \tag{3.11}$$

$$\beta_{qq} = \beta^0_{qq} + \sum_{i<j}^{N} \tau_{iq}\tau_{jl}X_{ij},\ \forall q \in \{1,\dots,Q\}, \tag{3.12}$$

$$\gamma_{ql} = \gamma^0_{ql} + \sum_{i\neq j}^{N} \tau_{iq}\tau_{jl}(m - X_{ij}),\ \forall q \neq l \in \{1,\dots,Q\}, \tag{3.13}$$

$$\gamma_{qq} = \gamma^0_{qq} + \sum_{i<j}^{N} \tau_{iq}\tau_{jl}(m - X_{ij}), \forall q \in \{1,\dots,Q\}. \tag{3.14}$$

Proof of (3.10). According to (3.2), we have

$$\begin{aligned} J(q(Z,\alpha,\pi)) &= \sum_Z \int\int q(Z,\alpha,\pi)\log\frac{\mathbb{P}(X,Z,\alpha,\pi)}{q(Z,\alpha,\pi)}\,d\alpha\,d\pi \\ &= \sum_Z \int\int (q(Z,\alpha,\pi)\log\mathbb{P}(X,Z,\alpha,\pi) - q(Z,\alpha,\pi)\log q(Z,\alpha,\pi))\,d\alpha\,d\pi \\ &= \mathbb{E}_{Z,\alpha,\pi}(\log\mathbb{P}(X,Z,\alpha,\pi)) - \mathbb{E}_{Z,\alpha,\pi}(\log q(Z,\alpha,\pi)) \\ &= \mathbb{E}_{Z,\pi}(\log\mathbb{P}(X|Z,\pi)) + \mathbb{E}_{Z,\alpha}(\log\mathbb{P}(Z|\alpha)) + \mathbb{E}_{\alpha}(\log\mathbb{P}(\alpha)) \\ &+ \mathbb{E}_{\pi}(\log\mathbb{P}(\pi)) - \sum_{i=1}^{N}\mathbb{E}_{Z_i}(\log q(Z_i)) - \mathbb{E}_{\alpha}(\log q(\alpha)) - \mathbb{E}_{\pi}(\log q(\pi)). \end{aligned} \tag{3.15}$$

By deriving (3.15) with respect to $q(\pi)$, and by taking this quantity equal to zero, we obtain :

$$
\begin{aligned}
\log q(\pi) =& \mathbb{E}_Z(\log \mathbb{P}(X|Z,\pi)) + \log \mathbb{P}(\pi) + \text{cst} \\
=& \sum_{i<j}^{N}\sum_{q,l}^{Q} \tau_{iq}\tau_{jl}\left(X_{ij}\log\pi_{ql} + (m - X_{ij})\log(1-\pi_{ql})\right) + \sum_{q\leq l}^{Q}((\beta_{ql}^0 - 1)\log\pi_{ql} \\
& + (\gamma_{ql}^0 - 1)\log(1-\pi_{ql})) + \text{cst} \\
=& \sum_{q<l}^{Q}\sum_{i\neq j}^{N} \tau_{iq}\tau_{jl}\left(X_{ij}\log\pi_{ql} + (m - X_{ij})\log(1-\pi_{ql})\right) + \sum_{q=1}^{Q}\sum_{i<j}^{N}\tau_{iq}\tau_{jq}(X_{ij}\log\pi_{qq} \\
& + (m - X_{ij})\log(1-\pi_{qq})) + \sum_{q\leq l}^{Q}\Big((\beta_{ql}^0 - 1)\log\pi_{ql} + (\gamma_{ql}^0 - 1)\log(1-\pi_{ql})\Big) \\
& + \text{cst} \\
=& \sum_{q<1}^{Q}\left((\beta_{ql}^0 - 1 + \sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}X_{ij})\log\pi_{ql} + (\gamma_{ql}^0 - 1 + \sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}(m - X_{ij}))\log(1-\pi_{ql})\right) \\
& + \sum_{q=1}^{Q}\left((\beta_{qq}^0 - 1 + \sum_{i<j}^{N}\tau_{iq}\tau_{jl}X_{ij})\log\pi_{qq} + (\gamma_{qq}^0 - 1 + \sum_{i<j}^{N}\tau_{iq}\tau_{jl}(m - X_{ij}))\log(1-\pi_{qq})\right) \\
& + \text{cst}. \qquad (3.16)
\end{aligned}
$$

By taking the exponential of (3.16), we obtain

$$
\begin{aligned}
q(\pi) &= e^{\sum_{q<1}^{Q}\left((\beta_{ql}^0 - 1 + \sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}X_{ij})\log\pi_{ql} + (\gamma_{ql}^0 - 1 + \sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}(m - X_{ij}))\log(1-\pi_{ql})\right)} \times \\
& \quad e^{\sum_{q=1}^{Q}\left((\beta_{qq}^0 - 1 + \sum_{i<j}^{N}\tau_{iq}\tau_{jl}X_{ij})\log\pi_{qq} + (\gamma_{qq}^0 - 1 + \sum_{i<j}^{N}\tau_{iq}\tau_{jl}(m - X_{ij}))\log(1-\pi_{qq})\right) + \text{cst}} \\
&= \text{cst}\prod_{q\leq l}^{Q}\pi_{ql}^{\beta_{ql}^0 - 1 + \sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}X_{ij}}(1-\pi_{ql})^{\gamma_{ql}^0 - 1 + \sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}(m - X_{ij})} \times \\
& \quad \prod_{q=1}^{Q}\pi_{ql}^{\beta_{qq}^0 - 1 + \sum_{i<j}^{N}\tau_{iq}\tau_{jl}X_{ij}}(1-\pi_{qq})^{\gamma_{qq}^0 - 1 + \sum_{i<j}^{N}\tau_{iq}\tau_{jl}(m - X_{ij})}.
\end{aligned}
$$

Thus, we obtain the product of Beta distributions (3.10). □

Now, we are interested in determining the approximation of the distribution $q(Z_i)$. By fixing $q(\alpha)$ and $q(\pi)$, and by maximizing the lower bound (3.2) with respect to $q(Z_i)$, we obtain that the approximation of the distribution $q(Z_i)$ is a multinomial distribution as follows

$$
q(Z_i) = \mathcal{M}(1, \tau_i = (\tau_{i1}, \ldots, \tau_{iQ})),
$$

where the parameter τ_{iq} denotes the probability of node i to belong to cluster q and can be expressed as follows

$$\tau_{iq} \propto e^{\left(\psi(n_q)-\psi(\sum_{r=1}^{Q} n_r)\right)} \prod_{i\neq j}^{N} \prod_{r=1}^{Q} e^{\left(\tau_{jr}\left(\log\binom{m}{X_{ij}}+(m\psi(\gamma_{qr})-m\psi(\beta_{qr}+\gamma_{qr})+X_{ij}(\psi(\beta_{qr})-\psi(\gamma_{qr}))\right)\right)}, \tag{3.17}$$

where ψ denotes the derivative of the logarithm of the gamma function (`digamma` in software `R`). The estimation of τ is obtained from (3.17) by iterating a fixed point algorithm until convergence.

Now, we introduce a theorem that we will use later in the proof.

Theorem 3.4.1. *If* $Y \sim$ `Beta`(a, b) *then* $\mathbb{E}_Y(\log Y) = \psi(a) - \psi(a+b)$ *and* $\mathbb{E}_Y(\log(1-Y)) = \psi(b) - \psi(a+b)$.

Proof of (3.17). According to (3.2), we have

$$\begin{aligned}
J(q(Z,\alpha,\pi)) &= \sum_{Z} \int\int q(Z,\alpha,\pi) \log \frac{\mathbb{P}(X,Z,\alpha,\pi)}{q(Z,\alpha,\pi)}\, d\alpha\, d\pi \\
&= \sum_{Z} \int\int (q(Z,\alpha,\pi)\log \mathbb{P}(X,Z,\alpha,\pi) - q(Z,\alpha,\pi)\log q(Z,\alpha,\pi))\, d\alpha\, d\pi \\
&= \mathbb{E}_{Z,\alpha,\pi}(\log \mathbb{P}(X,Z,\alpha,\pi)) - \mathbb{E}_{Z,\alpha,\pi}(\log q(Z,\alpha,\pi)) \\
&= \mathbb{E}_{Z,\pi}(\log \mathbb{P}(X|Z,\pi)) + \mathbb{E}_{Z,\alpha}(\log \mathbb{P}(Z|\alpha)) + \mathbb{E}_{\alpha}(\log \mathbb{P}(\alpha)) \\
&\quad + \mathbb{E}_{\pi}(\log \mathbb{P}(\pi)) - \sum_{i=1}^{N} \mathbb{E}_{Z_i}(\log q(Z_i)) - \mathbb{E}_{\alpha}(\log q(\alpha)) - \mathbb{E}_{\pi}(\log q(\pi)) \\
&= \sum_{i\leq j}^{N} \sum_{q,l}^{Q} \tau_{iq}\tau_{jl} \left(\log \binom{m}{X_{ij}} + X_{ij} \log \pi_{ql} + (m - X_{ij}) \log(1-\pi_{ql}) \right) \\
&\quad + \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \log \alpha_q + \mathbb{E}_{\alpha}(\log \mathbb{P}(\alpha)) + \mathbb{E}_{\pi}(\log \mathbb{P}(\pi)) - \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \log \tau_{iq} \\
&\quad - \mathbb{E}_{\alpha}(\log q(\alpha)) - \mathbb{E}_{\pi}(\log q(\pi)).
\end{aligned}$$

Since $\alpha_q \sim$ `Dir`(n_q) then $\alpha_q \sim$ `Beta`$(n_q, \sum_{q=1}^{Q} n_q - n_q)$. According to theorem 3.4.1, we have

$$\begin{aligned}
\mathbb{E}_{\alpha}(\log \alpha_q) &= \psi(n_q) - \psi(n_q + \sum_{q=1}^{Q} n_q - n_q) \\
&= \psi(n_q) - \psi(\sum_{q=1}^{Q} n_q).
\end{aligned}$$

However, since $\pi_{ql} \sim \texttt{Beta}(\beta_{ql}, \gamma_{ql})$, we have

$$\mathbb{E}_\pi(\log \pi_{ql}) = \psi(\beta_{ql}) - \psi(\beta_{ql} + \gamma_{ql})$$

and

$$\mathbb{E}_\pi(\log(1 - \pi_{ql})) = \psi(\gamma_{ql}) - \psi(\beta_{ql} + \gamma_{ql}).$$

Thus, the lower bound can be expressed as follows

$$\begin{aligned} J(q(Z,\alpha,\pi)) = & \sum_{i\leq j}^{N}\sum_{q,l}^{Q} \tau_{iq}\tau_{jl}(X_{ij}(\psi(\beta_{ql})-\psi(\beta_{ql}+\gamma_{ql}))+(m-X_{ij})(\psi(\gamma_{ql})-\psi(\beta_{ql}+\gamma_{ql}))) \\ & + \sum_{i\leq j}^{N}\sum_{q,l}^{Q} \tau_{iq}\tau_{jl}\left(\log\binom{m}{X_{ij}}\right) + \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\left(\psi(n_q) - \psi(\sum_{q=1}^{Q} n_q)\right) \\ & - \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\log\tau_{iq} + \mathbb{E}_\alpha(\log \mathbb{P}(\alpha)) + \mathbb{E}_\pi(\log \mathbb{P}(\pi)) - \mathbb{E}_\alpha(\log q(\alpha)) \\ & - \mathbb{E}_\pi(\log q(\pi)). \end{aligned} \tag{3.18}$$

By deriving (3.18) with respect to τ_i and by taking this quantity equal to zero, we obtain :

$$\begin{aligned} \log \tau_{iq} = & \sum_{i\neq j}^{N}\sum_{r=1}^{Q}\left(\tau_{jr}\left(\log\binom{m}{X_{ij}} + (m\psi(\gamma_{qr}) - m\psi(\beta_{qr}+\gamma_{qr})) + X_{ij}(\psi(\beta_{qr}) - \psi(\gamma_{qr}))\right)\right) \\ & + \left(\psi(n_q) - \psi(\sum_{q=1}^{Q} n_q)\right) + \text{cst}. \end{aligned}$$

By taking the exponential, we obtain (3.17).

□

According to (3.18) and using (3.6) and (3.10), we can express the lower bound (3.2) in an explicit form as follows

$$\begin{aligned} J(q(Z,\alpha,\pi)) \quad = \quad & \log\left\{\frac{\Gamma(\sum_q n_q^0)\prod_q \Gamma(n_q)}{\Gamma(\sum_q n_q)\prod_q \Gamma(n_q^0)}\right\} + \sum_{q\leq l}\log\left\{\frac{\Gamma(\beta_{ql}^0+\gamma_{ql}^0)+\Gamma(\beta_{ql})+\Gamma(\gamma_{ql})}{\Gamma(\beta_{ql}+\gamma_{ql})+\Gamma(\beta_{ql}^0)+\Gamma(\gamma_{ql}^0)}\right\} \\ & - \sum_i\sum_q \tau_{iq}\log\tau_{iq} + \sum_{i\leq j}^{N}\sum_{q,l}^{Q}\tau_{iq}\tau_{jl}\left(\log\binom{m}{X_{ij}}\right). \end{aligned} \tag{3.19}$$

Recall that $\Gamma(.)$ denotes the gamma distribution.

Proof.

$$
\begin{aligned}
J(q(Z,\alpha,\pi)) = &\sum_{i\leq j}^{N}\sum_{q,l}^{Q}\tau_{iq}\tau_{jl}(X_{ij}(\psi(\beta_{ql})-\psi(\beta_{ql}+\gamma_{ql}))+(m-X_{ij})(\psi(\gamma_{ql})-\psi(\beta_{ql}+\gamma_{ql})))\\
&+\sum_{i\leq j}^{N}\sum_{q,l}^{Q}\tau_{iq}\tau_{jl}\left(\log\binom{m}{X_{ij}}\right)+\sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\left(\psi(n_q)-\psi(\sum_{q=1}^{Q}n_q)\right)+\log\Gamma(\sum_{q=1}^{Q}n_q^0)\\
&+\sum_{q=1}^{Q}\left(n_q^0-1\right)\left(\psi(n_q)-\psi(\sum_{q=1}^{Q}n_q)\right)-\sum_{q=1}^{Q}\log\Gamma(n_q^0)+\sum_{q\leq l}^{Q}(\log\Gamma(\beta_{ql}^0+\gamma_{ql}^0)\\
&-\log\Gamma(\beta_{ql}^0)-\log\Gamma(\gamma_{ql}^0)+(\beta_{ql}^0-1)(\psi(\beta_{ql})-\psi(\beta_{ql}+\gamma_{ql}))\\
&+(\gamma_{ql}^0-1)(\psi(\gamma_{ql})-\psi(\beta_{ql}+\gamma_{ql})))-\sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\log\tau_{iq}-\log\Gamma(\sum_{q=1}^{Q}n_q)\\
&+\sum_{q=1}^{Q}\log\Gamma(n_q)-\sum_{q=1}^{Q}(n_q-1)\left(\psi(n_q)-\psi(\sum_{q=1}^{Q}n_q)\right)-\sum_{q\leq l}^{Q}(\log\Gamma(\beta_{ql}+\gamma_{ql})\\
&-\log\Gamma(\beta_{ql})-\log\Gamma(\gamma_{ql})+(\beta_{ql}-1)(\psi(\beta_{ql})-\psi(\beta_{ql}+\gamma_{ql}))\\
&+(\gamma_{ql}-1)(\psi(\gamma_{ql})-\psi(\beta_{ql}+\gamma_{ql})))\\
=&\sum_{i\leq j}^{N}\sum_{q,l}^{Q}\tau_{iq}\tau_{jl}\left(\log\binom{m}{X_{ij}}\right)+\sum_{q<l}^{Q}\Bigg((\beta_{ql}^0-\beta_{ql}+\sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}X_{ij})(\psi(\beta_{ql})\\
&-\psi(\beta_{ql}+\gamma_{ql}))+(\gamma_{ql}^0-\gamma_{ql}+\sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}(m-X_{ij}))(\psi(\gamma_{ql})-\psi(\beta_{ql}+\gamma_{ql}))\Bigg)\\
&+\sum_{q=1}^{Q}\Bigg((\beta_{qq}^0-\beta_{qq}+\sum_{i\neq j}^{N}\tau_{iq}\tau_{jq}X_{ij})(\psi(\beta_{qq})-\psi(\beta_{qq}+\gamma_{qq}))\\
&+(\gamma_{qq}^0-\gamma_{qq}+\sum_{i\neq j}^{N}\tau_{iq}\tau_{jq}(m-X_{ij}))(\psi(\gamma_{qq})-\psi(\beta_{qq}+\gamma_{qq}))\Bigg)\\
&+\sum_{q=1}^{Q}\left((n_q^0-n_q+\sum_{i=1}^{N}\tau_{iq})(\psi(n_q)-\psi(\sum_{l}^{Q}n_l))\right)\\
&+\log\left\{\frac{\Gamma(\sum_q n_q^0)\prod_q\Gamma(n_q)}{\Gamma(\sum_q n_q)\prod_q\Gamma(n_q^0)}\right\}+\sum_{q\leq l}\log\left\{\frac{\Gamma(\beta_{ql}^0+\gamma_{ql}^0)+\Gamma(\beta_{ql})+\Gamma(\gamma_{ql})}{\Gamma(\beta_{ql}+\gamma_{ql})+\Gamma(\beta_{ql}^0)+\Gamma(\gamma_{ql}^0)}\right\}\\
&-\sum_{i}\sum_{q}\tau_{iq}\log\tau_{iq}.
\end{aligned}
$$

Since we have :

— $n_q = n_q^0 + \sum_{i\neq j}^{N}\tau_{iq} \quad \forall q \in \{1,\dots,Q\}$

— $\beta_{ql} = \beta_{ql}^0 + \sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}X_{ij} \quad \forall q,l \in \{1,\dots,Q\}$

— $\beta_{qq} = \beta^0_{qq} + \sum_{i \neq j}^{N} \tau_{iq}\tau_{jq}X_{ij} \quad \forall q \in \{1, \ldots, Q\}$
— $\gamma_{ql} = \gamma^0_{ql} + \sum_{i \neq j}^{N} \tau_{iq}\tau_{jl}(m - X_{ij}) \quad \forall q, l \in \{1, \ldots, Q\}$
— $\gamma_{qq} = \gamma^0_{qq} + \sum_{i \neq j}^{N} \tau_{iq}\tau_{jq}(m - X_{ij}) \quad \forall q \in \{1, \ldots, Q\}$

Then, the equality (3.19) is reached. □

3.4.1 Variational Bayesian algorithm

We introduce here the algorithm of resolution of the model (see algorithm 4). We denote by t the current index for iterations in the algorithm and by ε a fixed threshold of convergence.

3.5 Model Selection

So far, we computed the approximate posterior distribution of all the model parameters and latent variables, given the observed data and the number of clusters Q. In this section, we are interested in determining the optimal number of clusters $\hat{Q}$ in the network.

The Bayesian framework provides a way of model selection. This framework estimates a probability distribution over a set of models, and the prediction is done by averaging over the ensemble of models. So, we develop a criterion based on a Bayesian approximation of the integrated observed data log-likelihood.

In the literature, there were only two model selection criteria developed to estimate the optimal number of clusters in `SBM` model. The integrated classification likelihood (`ICL`) developed by Biernacki et al. [2000] and Daudin et al. [2008] and the integrated likelihood variational Bayes (`ILvb`) developed by Latouche et al. [2012], which relies on a variational Bayesian approximation of the integrated observed data log likelihood. The `ICL` criterion aims at selecting the optimal number of clusters $\hat{Q}$ which maximizes the integrated observed-data log-likelihood, given a grid of values $\{1, \ldots, Q_{\max}\}$. The integrated observed-data log-likelihood can be expressed as follows

$$\log \mathbb{P}(X|Q) = \log \left\{ \sum_{Z} \int \int \mathbb{P}(X, Z, \alpha, \pi|Q) \mathrm{d}\alpha \mathrm{d}\pi \right\}. \tag{3.20}$$

This equation does not have an analytical expression. Indeed, it is intractable since it requires an integration over the model parameters α and π and the latent variables, for each value of Q. Since the variational framework offers elements of solution and makes it possible to tackle these problems simultaneously, we propose to use the integrated likelihood variational Bayes approach. So we replace the integrated observed-data log-likelihood (3.20) with its variational Bayesian

Algorithm 4 Variational Bayesian Expectation Maximization algorithm for inference in Binomial `SBM`

Initialization : Initialize τ^0 with a hierarchical algorithm based on the classical Ward distance by considering the Euclidean distance defined by $\text{dist}(i,j) = \sum_{m=1}^{n}(X_{im} - X_{jm})^2$. Initialize the vector n^0 by taking $\forall q, n_q^0 = \frac{1}{2}$ and the matrices β^0 and γ^0 by taking $\forall q, l, \beta_{ql}^0 = \gamma_{ql}^0 = \frac{1}{2}$ and give a random initialization value of the error rate *eps*.

1: Update the parameters n, β and γ iteratively (Bayes M-step)

$$n_q^{(t+1)} = n_q^0 + \sum_{i=1}^{N} \tau_{iq}^{(t)},\ \forall q \in \{1,\dots,Q\}.$$

$$\beta_{ql}^{(t+1)} = \beta_{ql}^0 + \sum_{i \neq j}^{N} \tau_{iq}^{(t)} \tau_{jl}^{(t)} X_{ij},\ \forall q \neq l \in \{1,\dots,Q\}.$$

$$\beta_{qq}^{(t+1)} = \beta_{qq}^0 + \sum_{i < j}^{N} \tau_{iq}^{(t)} \tau_{jl}^{(t)} X_{ij},\ \forall q \in \{1,\dots,Q\}.$$

$$\gamma_{ql}^{(t+1)} = \gamma_{ql}^0 + \sum_{i,j}^{N} \tau_{iq}^{(t)} \tau_{jl}^{(t)} (m - X_{ij}),\ \forall q \neq l \in \{1,\dots,Q\}.$$

$$\gamma_{qq}^{(t+1)} = \gamma_{qq}^0 + \sum_{i < j}^{N} \tau_{iq}^{(t)} \tau_{jl}^{(t)} (m - X_{ij}),\ \forall q \in \{1,\dots,Q\}.$$

2: Update the parameters τ iteratively (Bayes E-step)
while $|\tau^{new} - \tau^{old}| > eps$ **do**

$$\tau_{iq}^{new(t+1)} = e^{((\psi(n_q^{(t+1)}) - \psi(\sum_{r=1}^{Q} n_r^{(t+1)}))\times}$$
$$\prod_{i \neq j}^{N} \prod_{r=1}^{Q} e^{\left(\tau_{jr}^{old(t)} \left(\log\binom{m}{X_{ij}} + (m\psi(\gamma_{qr}^{(t+1)}) - m\psi(\beta_{qr}^{(t+1)} + \gamma_{qr}^{(t+1)}) + X_{ij}(\psi(\beta_{qr}^{(t+1)}) - \psi(\gamma_{qr}^{(t+1)})))\right)\right)}.$$

end while
$\tau^{(t+1)} \longrightarrow \tau^{new(t+1)}$.

3: Calculate the lower bound iteratively

$$J^{(t+1)} = \log\left\{\frac{\Gamma(\sum_q n_q^0)\prod_q \Gamma(n_q^{(t+1)})}{\Gamma(\sum_q n_q^{(t+1)})\prod_q \Gamma(n_q^0)}\right\} + \sum_{q \leq l} \log\left\{\frac{\Gamma(\beta_{ql}^0 + \gamma_{ql}^0) + \Gamma(\beta_{ql}^{(t+1)}) + \Gamma(\gamma_{ql}^{(t+1)})}{\Gamma(\beta_{ql}^{(t+1)} + \gamma_{ql}^{(t+1)}) + \Gamma(\beta_{ql}^0) + \Gamma(\gamma_{ql}^0)}\right\}$$
$$- \sum_i \sum_q \tau_{iq}^{(t+1)} \log \tau_{iq}^{(t+1)} + \sum_{i \leq j}^{N} \sum_{q,l}^{Q} \tau_{iq}^{(t+1)} \tau_{jl}^{(t+1)} \left(\log\binom{m}{X_{ij}}\right).$$

4: Repeat Step 1 and 2 until $\|J^{(t+1)} - J^{(t)}\| < \varepsilon$.

approximation. Given the value of Q, we are interested in maximizing the lower bound (3.19) with respect to $q(.)$. Recall that the lower bound is of the form

$$\begin{aligned} J_Q(q(Z,\alpha,\pi)) &= \sum_Z \int\int q(Z,\alpha,\pi)\log\frac{\mathbb{P}(X,Z,\alpha,\pi|Q)}{q(Z,\alpha,\pi)} \\ &= \log\left\{\frac{\Gamma(\sum_q n_q^0)\prod_q\Gamma(n_q)}{\Gamma(\sum_q n_q)\prod_q\Gamma(n_q^0)}\right\}+\sum_{q\leq l}\log\left\{\frac{\Gamma(\beta_{ql}^0+\gamma_{ql}^0)+\Gamma(\beta_{ql})+\Gamma(\gamma_{ql})}{\Gamma(\beta_{ql}+\gamma_{ql})+\Gamma(\beta_{ql}^0)+\Gamma(\gamma_{ql}^0)}\right\} \\ &\quad -\sum_i\sum_q \tau_{iq}\log\tau_{iq}+\sum_{i\leq j}^{N}\sum_{q,l}^{Q}\tau_{iq}\tau_{jl}\left(\log\binom{m}{X_{ij}}\right). \end{aligned}$$

Note that maximizing the lower bound is equivalent to minimizing the Kullback-Leibler divergence between $q(.)$ and the unknown posterior distribution. After convergence of the algorithm, although the Kullback-Leibler divergence distance (3.3) can not be computed analytically. We expect it to be close to zero. Therefore, we can use the lower bound as an approximation of $\log\mathbb{P}(X|Q)$. This leads to a new criterion for `SBM` called `ILvb`.

The `ILvb` can be expressed as follows

$$\begin{aligned} \texttt{ILvb} &= \log\left\{\frac{\Gamma(\sum_q n_q^0)\prod_q\Gamma(n_q)}{\Gamma(\sum_q n_q)\prod_q\Gamma(n_q^0)}\right\}+\sum_{q\leq l}\log\left\{\frac{\Gamma(\beta_{ql}^0+\gamma_{ql}^0)+\Gamma(\beta_{ql})+\Gamma(\gamma_{ql})}{\Gamma(\beta_{ql}+\gamma_{ql})+\Gamma(\beta_{ql}^0)+\Gamma(\gamma_{ql}^0)}\right\} \\ &\quad -\sum_i\sum_q \hat{\tau}_{iq}\log\hat{\tau}_{iq}+\sum_{i\leq j}^{N}\sum_{q,l}^{Q}\hat{\tau}_{iq}\hat{\tau}_{jl}\left(\log\binom{m}{X_{ij}}\right). \end{aligned}$$

To calculate the optimal number of clusters $\hat{Q}$ in the network, we run the variational Bayesian `EM` algorithm for different values of Q. Then we choose for $\hat{Q}$ the value that maximizes `ILvb`.

3.6 Numerical Experiments

The purpose of this section is to illustrate numerically the main features of the proposed method as well as to compare it to the binomial `SBM` method developed in the previous chapter. We resume first the simulated data examples and then the three applications used in the previous chapter. Then, we apply the proposed method to show numerically the obtained results and to compare them to those obtained in the previous chapter by using the binomial `SBM`.

3.6.1 Simulated data

We resume here the first simulated data example. Recall that the simulated network has $n = 20$ vertices, a fixed number of clusters Q chosen equal to three

and that the parameters used in this simulation are :

$$\bar{\alpha} = (0.2, 0.5, 0.3)$$

and

$$\bar{\pi} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}.$$

Furthermore, we present here the graph associated to the simulated network which is already introduced in the previous chapter. This graph is built using `Gephi` software with the layout algorithm "Force Atlas". We can show in Figure 3.1 the structure of the simulated data graph. There are three apparent communities.

By applying our algorithm implemented in `R` programming language, we obtain that the vertices of the network are grouped into three clusters as shown in Table 3.1.

Clusters	**Vertices**
Red	3 8 9 14 20
Blue	4 5 6 7 10 11 12 13 15 17
Green	1 2 16 18 19

TABLE 3.1 – Grouping first simulated network vertices into clusters using the variational Bayesian `SBM`

Table 3.1 shows clearly that the nodes of the first simulated graph are split into three clusters which are the same as the three clusters shown in the Figure 3.1. That confirms the effectiveness of our method. The time of convergence of the algorithm is 0.12 second (CPU Corel3 - 4GB RAM) which is so satisfying.

By comparing Table 3.1 with Table 2.1, we can conclude that the two methods give the same results. Thus, the vertices of the network are grouped into the same three clusters by using the the binomial `SBM` or the binomial variational Bayesian `SBM`.

Now, by calculating the `ARI` defined in the previous chapter between the estimated clustering results obtained by the binomial `SBM` and those obtained by the proposed method, we obtain `ARI`=1. This means that the two partitions of the nodes agree perfectly.

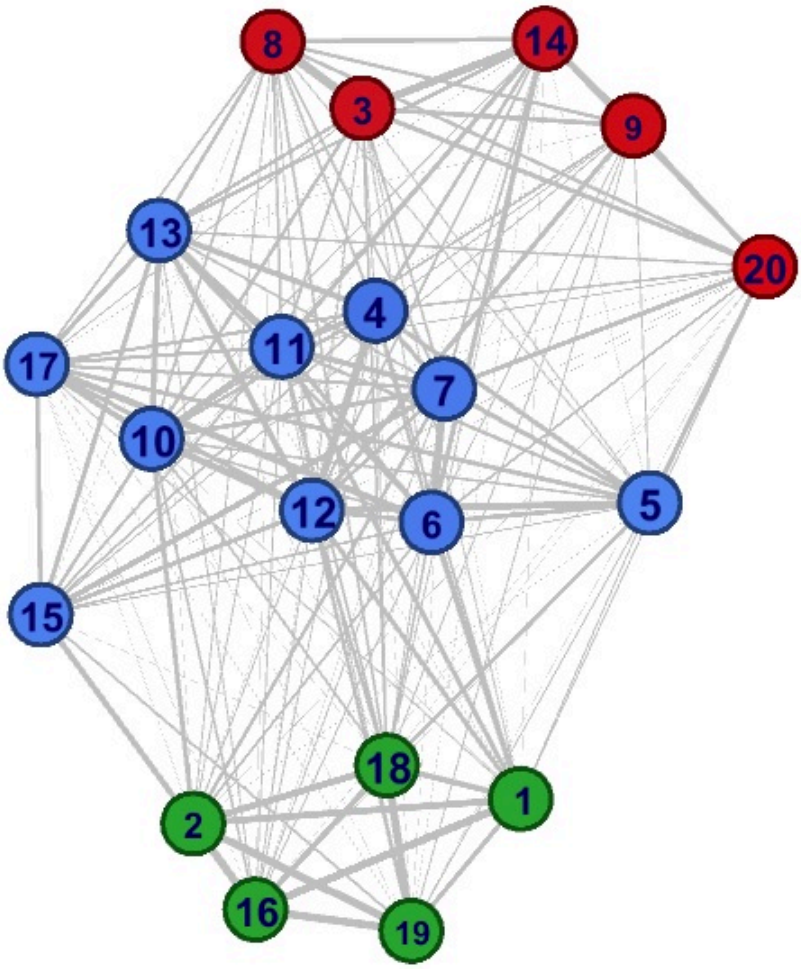

FIGURE 3.1 – First simulated data graph visualization with `Gephi`.

3.6.2 Co-citation networks

We resume here the two co-citation networks developed in the previous chapter. Then, we apply our proposed method to show numerically the clustering results and to compare these results to those obtained in the previous chapter by using the binomial `SBM`.

Twitter network's data

We resume here the twitter network's data. Recall that this data consists of 154 tweets and 21 terms and has the form of a tweet-by-term matrix. Note that this data is available online at http://www.rdatamining.com/data. As mentioned

in the previous chapter, we transform the tweet-by-term matrix into a term-by-term matrix based on the co-occurrence of term in the same tweets. The network associated to the matrix is an undirected co-citation network consisting of 21 vertices and 130 edges. Each vertex represents a term and there is an edge between a pair of terms if they co-occur together a least one time in the tweets.

We present the graph associated to the twitter network which is already introduced in the previous chapter. This graph is built using `Gephi` software with the layout algorithm "Force Atlas".

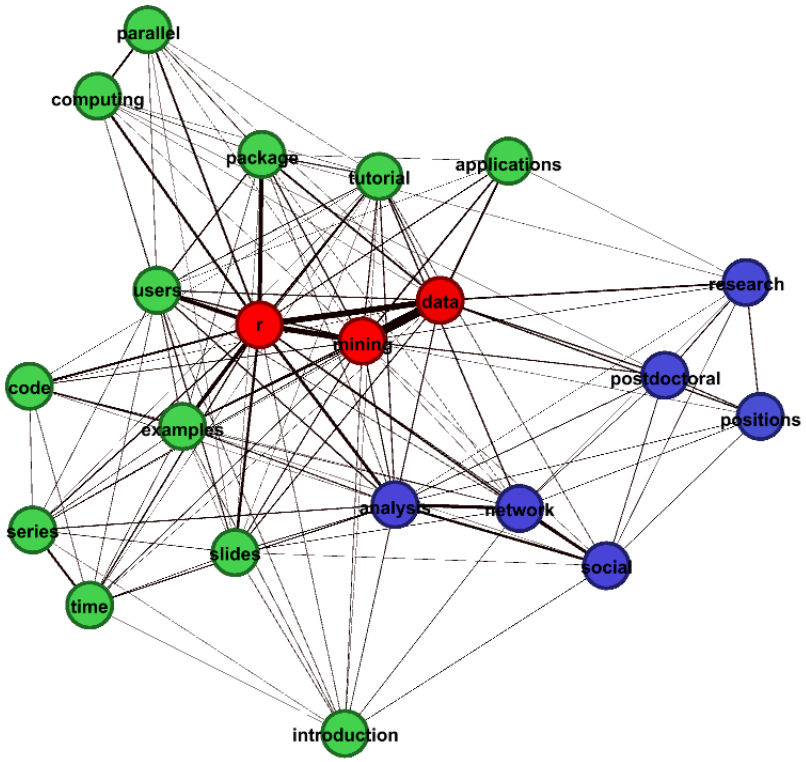

FIGURE 3.2 – Twitter network of terms visualization with `Gephi`.

By applying our algorithm implemented in software `R`, we obtain that the terms of the network are grouped into three groups as shown in the following Table 3.2.

We can show in Table 3.2 the classification of the twitter network's terms into clusters. Thus, the terms of each cluster are often cited together in the tweets.

Furthermore, by comparing the results obtained in Table 3.2 to those obtained in the previous chapter by using the binomial `SBM`, we can notice that the two methods give the same results. Thus, the vertices of the network are grouped into

Clusters	vertices
Red	r data mining
Blue	research postdoctoral positions analysis social network
Green	parallel computing time series code examples slides applications package users tutorial introduction

TABLE 3.2 – Grouping the terms of the twitter network into clusters using the variational Bayesian binomial SBM.

the same three clusters by using the the binomial SBM or the binomial variational Bayesian SBM.

Now, by calculating the ARI defined in the previous chapter between the estimated clustering results obtained by the binomial SBM and those obtained by the proposed method, we obtain ARI=1. This means that the two partitions of the nodes agree perfectly.

Reuters-21578 Network's data

We resume here the Reuters-21578 network's data. The data is developed and detailed in the previous chapter. Recall that the data is a corpus of 20 documents available in the package tm of the software R under the name of crude and that we have built in the previous chapter a term-by-document matrix of this corpus by doing a text mining treatment. This obtained term-by-document matrix is of size 20×21 and consists of 20 documents and 21 terms.

We transform the term-by-document matrix into a term-by-term matrix. The network associated to this matrix is an undirected network of 21 vertices and 97 edges, where each vertex is a term and there is an edge between a pair of terms if they co-occur together at least one time in the documents.

The graph associated with this network is visualized in Figure 3.3 using Gephi software with the layout algorithm Force Atlas.

By applying our algorithm implemented in software R, we obtain that the terms of the network are grouped into four clusters as shown in Table 3.3.

Clusters	Vertices
Red	oil
Blue	mln bpd month sources production saudi market opec prices
Green	billion budget riyals government economics indonesia report
Cyan	exchange nymex futures Kuwait

TABLE 3.3 – Grouping the terms of the network of terms of the Reuters-21578 corpus into clusters using the variational Bayesian binomial SBM.

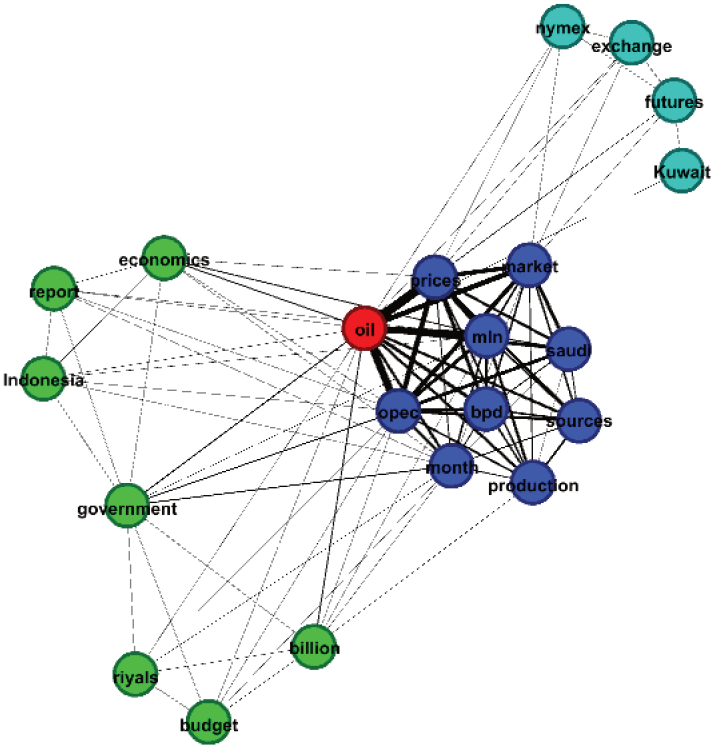

FIGURE 3.3 – Network of terms of the the Reuters-21578 corpus visualization with `Gephi`.

Table 3.3 shows the distribution of the network's terms into groups which means that the terms of each group are frequently co-occuring together in the documents.

In the following, we compare in Figure 3.4 the grouping results obtained by the binomial variational Bayesian `SBM` to those obtained by the binomial `SBM`. Recall that `bSBM` means the binomial `SBM` while `bVBSBM` means the binomial variational Bayesian `SBM`.

Figure 3.4 shows a small difference between the results obtained by the variational Bayesian binomial `SBM` and those obtained by the binomial `SBM`.

Now, by calculating the `ARI` between the estimated clustering results obtained by the binomial `SBM` and those obtained by the proposed method, we obtain `ARI`=0.8. This means a high agreement between the two partitions of the nodes.

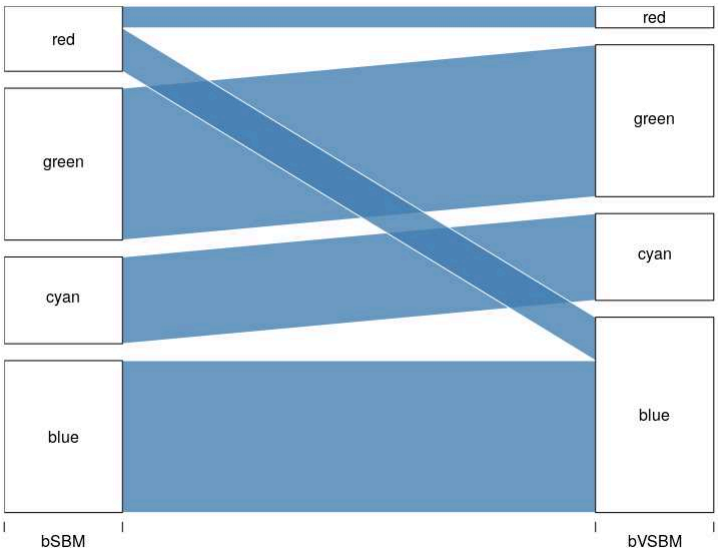

FIGURE 3.4 – Comparison of the clustering results of the Reuter network obtained by the binomial SBM and the binomial variational Bayesian binomial SBM.

3.6.3 Social network : a benchmark dataset

We resume here the deep South network developed in the previous chapter. Then, we apply our proposed method to show numerically the clustering results and to compare these results to those obtained in the previous chapter using the binomial SBM.

Deep South network

We resume here the deep South network developed in the previous chapter. Recall that the data is available in the package manet in software R under the name deepsouth http://cran.r-projet.org/web/packages/manet/manet.pdf and that it represents the participation of 18 Southern women in a series of 14 informal social events over a nine-month period. The data has the form of an event-by-women matrix. We transformed in the previous chapter this matrix into a women-by-women matrix by multiplying the data matrix by its transpose. The obtained network is an undirected network of 18 vertices and 139 edges, where each vertex represents a Southern women among the 18. There is an edge between a pair of women if they participate together in one of the 14 events a least.

The graph associated with the obtained network is visualized in Figure 3.5 using Gephi software with the layout algorithm Force Atlas.

By applying our algorithm implemented in software R, we obtain that the

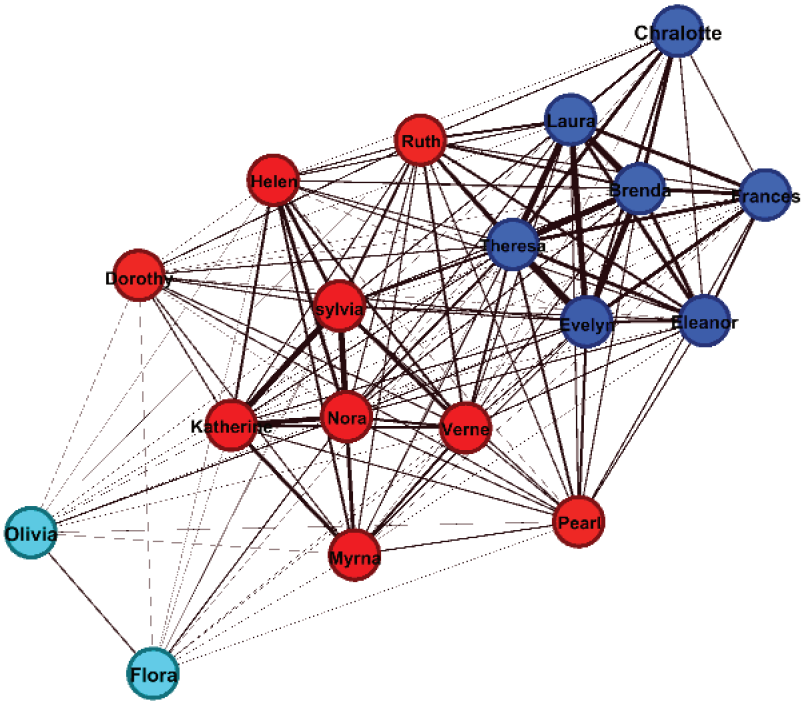

FIGURE 3.5 – Deep South network visualization with `Gephi`.

eighteen women are grouped into three clusters as shown in Table 3.4.

Clusters	Vertices
Cyan	Olivia Flora
Blue	Evelyn Laura Theresa Brenda Charlotte Frances Eleanor
Red	Verne Myrna katherine Sylvia Nora Helen Pearl Ruth Dorothy

TABLE 3.4 – Grouping the women of deep South network into clusters.

In table 3.4, each cluster represents the women which are frequently met together in the informal social events.

In the following, we compare in Figure 3.6 the grouping results obtained by the binomial variational Bayesian `SBM` to those obtained by the binomial `SBM`. Recall that `bSBM` means the binomial `SBM` while `bVBSBM` means the binomial variational Bayesian `SBM`.

Figure 3.6 shows a small difference between the results obtained by using the binomial `SBM` and those obtained by using the variational Bayesian binomial `SBM`.

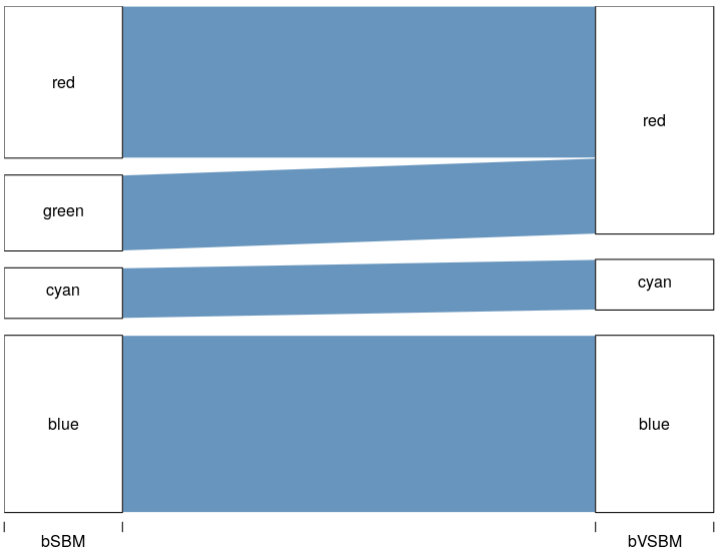

FIGURE 3.6 – Comparison of the clustering results of the deep South network obtained by the binomial SBM and the binomial variational Bayesian binomial SBM.

Now, by calculating the ARI between the estimated clustering results obtained by the binomial SBM and those obtained by the proposed method, we obtain ARI=0.73. This means a high agreement between the two partitions of the nodes.

3.7 Application : Co-citation networks in statistical text mining

We introduce in this section an application of the proposed method. Then, we compare the obtained results to those obtained by using the binomial SBM approach developed in the previous chapter.

Co-citation networks in statistical text mining

Migration interviews

We apply our model to analyze a corpus of interviews of migrant minors, from Sub-Sahara to the Mediterranean European coast[2]. About one hundred minors in migration accepted to answer to a semi-directed interview. Their testimonies were put into numeric texts. A pre-treatment of the digital corpus, in French, was

2. This corpus was constituted by N. Robin, Research Geographer (HDR), CEPED, UMR 196 (Paris Descartes-IRD), hosted at MIGRINTER (CNRS), UMR 7301.

done using tm package in R. In particular, lemmatization was done. No stemming was applied. Considering the list of nouns and adjectives, ranked by frequency in the whole corpus, a list of the first 25 most relevant words was established. The choice was made by N. Robin in agreement with her expertise of the topic. This list is[3] : "argent/money", "route/road", "voyage/travel", "famille/family", "Europe", "Gao", "avenir/future", "gens/people", "jour/day", "police", "Bordj", "passeport/passport", "contact", "camion/truck", "travail/work", "oncle/uncle", "difficile/difficult", "malien", "ami/friend", "passeur/smuggler", "parent", "transport", "Mali", "foyer/home", "projet/project". A very first analysis was done in (Louis and Robin [2016]). A broader analysis of this corpus through a larger list of words is in progress and will be the topic of a dedicated paper.

Here we do consider the text-document matrix associated with these 25 accurately chosen words. Each document is an interview. This matrix was then converted into a co-citation matrix (25×25 term-by-term matrix) : to each couple of words is associated the number of interviews where both words are jointly used. The network associated to the matrix is an undirected network of 25 vertices and 300 edges. Each vertex is a word. There is an edge between a couple of words if they co-occur together at least one time in the documents. The weight associated to this edge is the number of interviews were at least one co-citation occurs. The graph associated with the network is visualized in Figure 3.7 using `Gephi` software with the layout algorithm Force Atlas.

We present in Table 3.5 the assortativity coefficient, the average clustering coefficient and the density of migrants interview network.

Assortativity	**Average clustering coefficient**	**Density**
0.24	0.97	1

TABLE 3.5 – Global characteristics of migrants interview network's structure.

We can show in Table 3.5 that the assortativity coefficient is equal to 0.24 which is a positive value. That means that the terms presented in the documents of the corpus tends to occur with other terms that have equally high or equally low number of occurrence. The average clustering coefficient (transitivity) is equal to 0.97 which shows the completeness of the neighborhood of the vertices in the network. The density of the graph is equal to 1 which indicates that the graph of the network is dense. Note that the transitivity and the density value are close which means that the network is not highly clustered.

By applying our algorithm implemented in software `R`, we obtain that the words of the network are grouped into three groups as shown in the following Table 3.6.

3. English translation is here added for clarity. Capital letters were considered as small letters in the analysis.

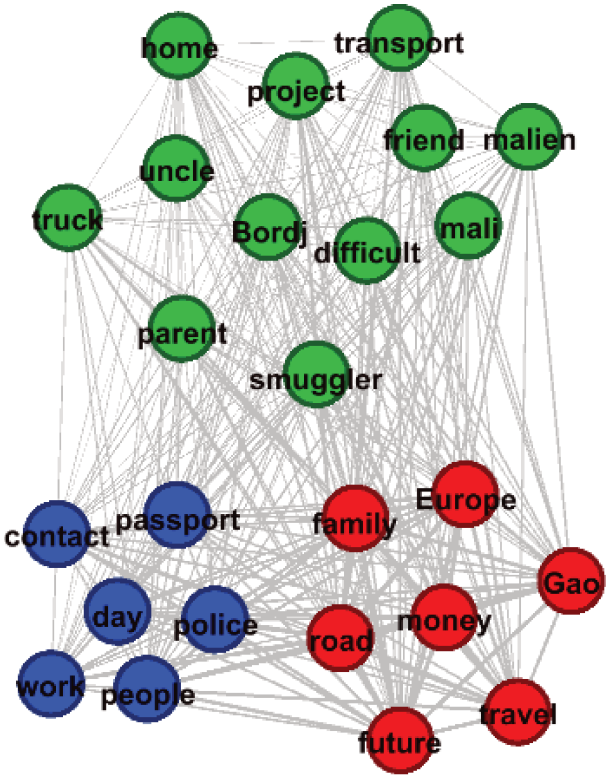

FIGURE 3.7 – Migrants interview network visualization with `Gephi`.

Clusters	Vertices
Red	money future Europe family Gao road travel
Blue	contact people day passeport police work
Green	friend Bordj truck difficult home mali malien uncle parent smuggler project transport

TABLE 3.6 – Grouping the words of migrants interview network into clusters.

There are three clusters found by the methodology in the Table 3.6.

The first one, whose nodes are colored red in Figure 3.7, is more related to the motivation of the migration : family is motivating the travel, on order to reach

Europe, where future and money are available. Gao is a central place where many resources are available, which make the migration possible.

The second cluster in blue are more obstacles to the migration : passport is needed, contacts/people too, part of the travel needs some days. You need to avoid the police.

The third cluster (colored in green) is more related to the travel's means : transportation, truck, smuggler. Uncle and parents are helping the migration. Bordj is a border city.

It is interesting to notice that two cities : Bordj and Gao are associated to different clusters. The works of the geographers confirm these two cities play different roles with respect to migrations according to available resources.

Clearly, this is a very first interpretation on a small number of highly quoted words. This approach needs to be implemented on the full corpus.

Now, by applying the binomial `SBM` method developed in the previous chapter, we obtain the same clustering results. Thus, the two method yields to the same results.

Introduction en Français du Chapitre 4

Ceci est une introduction en français du chapitre "Clustering in Attributed Weighted Nodes Network using Stochastic Block Model with Application to Electroencephalographic Data".

Introduction

Comprendre comment le cerveau humain permet l'écriture est un des enjeux de la psycholinguistique (Perret and Olive [2019]). Parmi les outils utilisés pour comprendre cette fonction cognitive, l'enregistrement de l'activité électrique du cerveau (`EEG`) offre de nombreux avantages. En particulier, cette méthode permet de suivre temporellement les différentes activités réalisées par le cerveau lors de l'écriture. Les travaux en psycholinguistique s'appuient sur des tâches dont l'objectif est à la fois de faire produire du comportement mesurable aux participants et d'avoir un contrôle expérimental sur ce qui est produit. Une tâche très souvent utilisée est la dénomination d'images. Le participant est assis devant un écran d'ordinateur et dispose d'une tablette graphique, d'un stylet et d'une feuille. Une image est présentée au participant (*e.g.*, une araignée) et ce dernier doit le plus rapidement possible et le plus correctement possible écrire à la main le nom de cette image.

Une des périodes d'intérêt pour comprendre les processus cognitifs impliqués dans la production écrite s'étend de la présentation de l'image jusqu'au premier mouvement du participant sur la tablette graphique. Ce temps est nommé latence d'initialisation. Il correspond à la durée mise par le cerveau pour réaliser les différentes activités nécessaires pour commencer à écrire le nom de l'image. Cela recouvre la perception visuelle, la reconnaissance de l'objet présentée, l'accès au mot *i.e.*, la récupération en mémoire des différentes lettres constitutives du mot à produire et la planification des gestes nécessaires à l'écriture. Chacune de ces étapes fait l'objet d'études afin de mieux comprendre comment elle est réalisée par le cerveau. Par exemple, comprendre comment un être humain récupère les lettres

constitutives d'un mot implique de se demander comment cette information est stockée en mémoire. Les enregistrements électro-encéphalographique (EEG) permettent d'avoir accès à chacune de ces périodes.

Grâce à un système d'électrodes posé à la surface du scalp du participant, l'activité électrique produite par le cerveau est enregistrée en continu durant la latence d'initialisation. Deux types de cellules constituent le cerveau. Le premier type correspond aux cellules gliales. Elles forment la cytostructure et vont jouer un rôle dans le soutien et la protection. Le second type, le plus connu, est le neurone. Il s'agit des cellules à l'origine de l'esprit humain et donc celles qui nous intéressent. Par échange d'ions entre l'intérieur et l'extérieur de la cellule, les neurones produisent une infime quantité d'électricité en continue et de manière aléatoire. Ce signal électrique joue un rôle important dans la communication des neurones entre eux. Toutefois, ce signal est tellement faible qu'il est impossible de l'enregistrer. Dans certaines situations, l'émission d'électricité de groupes de neurones se modifie et devient une activité synchronisée. Partant de la création d'une petite quantité d'électricité par chaque neurone de manière aléatoire, plusieurs centaines de neurones vont produire au même moment du signal électrique. Cet ensemble de signaux est moyenné et forme un dipôle de courant équivalent (DCE). Ce DCE, de l'ordre que quelques microvolts, est mesurable à la surface du scalp. Cela correspond à l'activité électrique enregistré en électroencéphalographie.

Une des situations à l'origine de la synchronisation de groupe de neurones est leur implication dans une des activités cognitives nécessaires à la production écrite. Plus précisément, des groupes de neurones sont dédiés à chacune des étapes de traitement cognitif nécessaires pour écrire le nom d'une image. De plus, ces groupes de neurones sont répartis à travers tout le cerveau. Par exemple, la perception visuelle implique des neurones présents dans le cortex occipital situé à l'arrière du cerveau alors que la planification motrice implique des neurones situés dans la zone fronto-pariétale gauche pour un participant droitier, en haut du crâne en aplomb de l'oreille.

Motivation

Ce chapitre vise au développement d'un modèle pour un outil d'aide à la spécification des différents traitements cognitifs réalisés par le cerveau lors de la préparation de l'écriture. Un participant a produit par écrit le nom de 120 images (Perret and Laganaro [2012]). L'activité EEG a été enregistrée à l'aide du système de 128 électrodes répartis à la surface du scalp (ActiveTwo Biosmemi EEG sytem, V.O.F. Amsterdam, Netherlands). Un ensemble de traitement du signal (Luck [2005] ; Michel el al. [2009]) a été réalisé afin d'obtenir un potentiel évoqué pour l'ensemble des données (Event-related Potential, ERP) à partir d'une bande passante de signal comprise entre 0.2 et 30Hz.

Comme décrit ci-dessus, le ERP regroupe l'activité enregistrée en continu des différents groupes de neurones dont l'activité synchronisée a créé un ECD. Autrement dit, la moyenne global correspond à une série d'ECD se succédant temporellement. Cela amène à faire une hypothèse en termes de conformation spatiale de l'activité électrique : il est possible de suivre précisément la période d'activité de chaque groupe de neurones en s'appuyant sur les changements de la répartition spatiale à travers le temps de l'activité électrique collectée à la surface du scalp. En effet, à un instant donné, une mesure d'intensité du courant électrique en microvolt est faite pour chacune des 128 électrodes. En associant toutes les électrodes, il est alors possible de décrire une configuration spatiale de l'activité électrique à un instant t, appelée topography or carte. Une topographie est alors le résultat de l'organisation spatiale du niveau d'intensité électrique des 128 électrodes les unes par rapport aux autres (voir figure B.1). Un ECD est généré durant une période temporelle précise, celle durant laquelle le groupe de neurones est synchronisé. Ainsi, il semble possible de segmenter l'ERP en une série de configurations spatiales stables de l'activité électrique, séparées par de brusques transitions (Michel el al. [2009]). Le travail du psycholinguiste est ensuite d'associer chaque configuration spatiale aux traitements cognitifs.

L'objectif de ce chapitre est de classifier les 128 électrodes pour chaque pas de temps. Les données consistent en 128 électrodes et 285 pas de temps et se présentent sous la forme d'une matrice électrodes-par-pas de temps. Pour chaque pas de temps T_i, nous transformons la matrice électrodes-par-T_i constituée des électrodes et de ce pas de temps en une matrice électrodes-par-électrodes de dimension 128×128. Le réseau considéré est alors constitué de 128 noeuds. Chaque électrode correspond à un noeud. De plus, chaque noeud est associé à un vecteur de poids représentant la différence absolue entre l'intensité du signal de l'électrode et celle des électrodes voisines. Le voisinage est défini par rapport aux positions des électrodes sur le bonnet. Ce sont les électrodes proches spatialement. L'intensité électrique pouvant être positive ou négative, un signe a été attribué pour chaque arête entre une paire d'électrodes. Ce signe est positif si la valence de l'intensité des deux noeuds est la même (+/+ ou -/-). Il est négatif si la valence est différente pour les deux noeuds (+/- ou -/+). Le réseau considéré est alors un réseau binaire ayant des poids associés aux noeuds. Afin de classifier ces noeuds, on a développé un modèle à blocs stochastiques. Une approche variationnelle est considérée pour estimer les paramètres du modèle ainsi que pour classifier ces noeuds. L'objectif est de regrouper les électrodes en cluster afin d'explorer les variations en termes d'intensité moyenne des clusters à travers le temps.

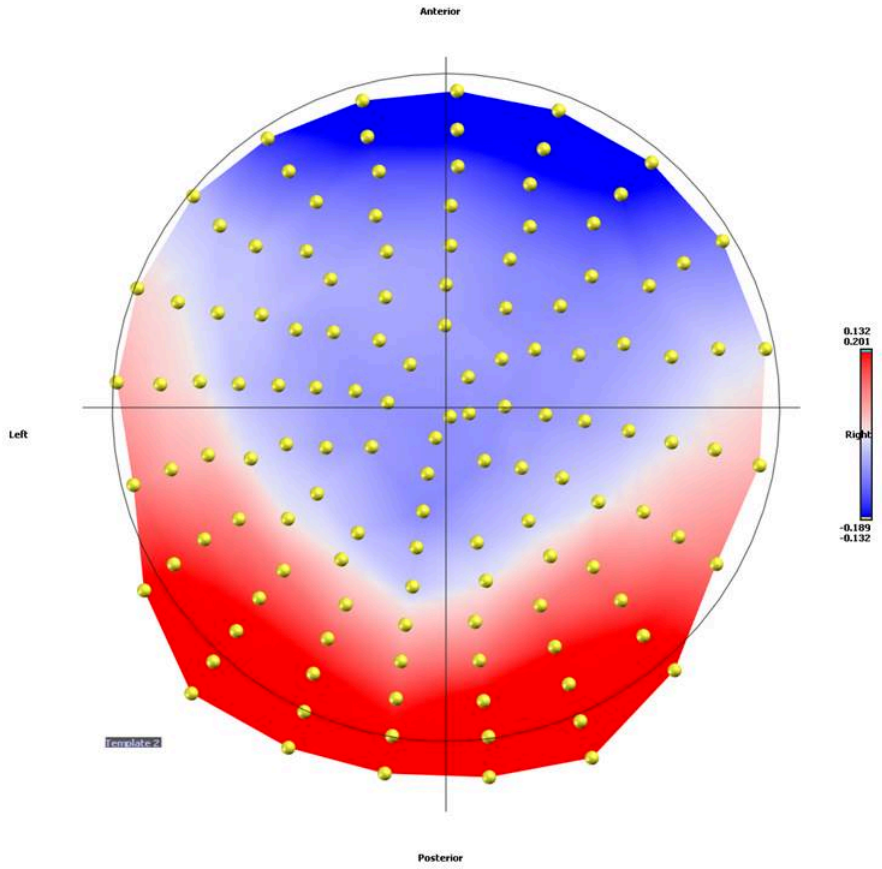

FIGURE B.1 – Exemple de topographie stable d'activité électrophysiologique.

Structure du Chapitre

Dans ce chapitre, nous définissons un réseau binaire non orienté avec des poids attribués aux noeuds dans la section 4.2. Nous définissons le modèle à blocs stochastiques proposé dans la section 4.3. Dans la section 4.4, nous réalisons une inférence variationnelle du modèle à blocs stochastiques proposé. En effet, dans la sous section 4.4, nous introduisons l'algorithme espérance maximisation variationnel pour estimer les paramètres de ce modèle. Dans la section 4.5, nous adoptons un critère de sélection du nombre de clusters qui s'adapte de manière optimale aux données. Enfin, nous introduisons dans la section 4.6 une application sur des données d'électro-encephalographique (`EEG`) afin de spécifier les différents traitement cognitifs réalisé par le cerveau humain lors de la préparation de l'écriture à partir de l'activité électrique produite par les neurones de ce cerveau et enregistré par l'électroencéphalogramme.

Chapitre 4

Clustering in Attributed Weighted Nodes Network using Stochastic Block Model with Application to Electroencephalographic Data

4.1 Introduction

The understanding of how human brain allows writing is one of the issues of psycholinguistics (Perret and Olive [2019]). Among the tools used to understand this cognitive function, the recording of brain electrical activity (electroencephalography, `EEG`) offers many advantages. In particular, this method makes it possible to follow temporally different activities performed by the brain during writing. The work in psycholinguistics is based on tasks, designed both to produce measurable behavior to participants and to have experimental control over what is produced. A very common task is the picture naming. The participant sits in front of a computer screen and has a graphic tablet, a stylus and a sheet. An drawing object is presented to the participant (*e.g.*, a spider). He/she has to handwrite as soon and as accurately as possible the name of this drawing. One of the periods of interest to understand the cognitive processes involved in handwritten production extends from the presentation of the picture to the first movement of the participant on the graphic tablet. This time is called initialization latency. It corresponds to the duration put by the brain to carry out the various activities necessary to start handwriting the name of the image. This covers the visual perception, the recognition of the object, the wordform access (*i.e.*, the recovery in memory of the different

letters constituting the word to produce) and the planning of the gestures necessary for handwriting. Each of these steps is studied to better understand how it is performed by the brain. For example, understanding how a participant retrieves the letters that make up a word implies asking how this information is stored in memory. Electroencephalographic (EEG) recordings provide access to each of these periods. The electrical activity produced by the brain is recorded continuously during the initialization latency due to an electrode system placed on participant's scalp. Two types of cells make up the brain. The first type corresponds to glial cells. They form the cytostructure and will play a role in support and protection. The second type, the best known, is the neuron. These are the cells at the origin of the human mind and therefore those that interest us. By ion exchange between the inside and outside of the cell, the neurons produce a tiny amount of electricity continuously and randomly. This electrical signal plays an important role in the communication of neurons with each other. However, this signal is so weak that it cannot be recorded. In some situations, the emission of electricity from neuronal groups changes and becomes a synchronized activity. Starting from the creation of a small amount of electricity by each neuron in a random manner, several hundred neurons will produce at the same time the electrical signal. This set of signals is averaged and forms an Equivalent Current Dipole (ECD). This ECD, of the order of a few microvolts, is measurable on the surface of the scalp. This corresponds to the electrical activity recorded in electroencephalography. One of the situations at the origin of the neuron group synchronization is their implication in one of the cognitive activities necessary for the handwritten picture naming task. Specifically, groups of neurons are dedicated to each of the cognitive processing steps required to handwrite the name of a picture. In addition, these groups of neurons are distributed throughout the brain. For example, visual perception involves neurons present in the occipital cortex located at the back of the brain while motor planning involves neurons located in the left frontal-parietal are for a right-handed participant, at the top of the skull in line with the hear.

4.1.1 Motivation

This chapter aims to develop a model useful for tool to help the specification of different cognitive treatments performed by the brain during the preparation of handwriting. A participant produced the name of 120 images in writing (Perret and Laganaro [2012]). EEG activity was recorded using the system of 128 electrodes distributed on the surface of the scalp (ActiveTwo Biosmemi EEG sytem, V.O.F. Amsterdam, Netherlands). A set of signal processing (Luck [2005] ; Michel el al. [2009]) was realized in order to obtain a potential evoked for the data (*Event-related Potential*, ERP) from a signal bandwidth between 0.2 and 30Hz.

As described above, the ERP groups continuously recorded activity of the dif-

ferent groups of neurons whose synchronized activity creates an ECD. In other words, the grand average corresponds to a series of consecutive ECDs. This leads to a hypothesis in terms of the spatial conformation of the electrical activity : it is possible to precisely follow the period of activity of each group of neurons by relying on changes in the spatial distribution over time of the electrical activity collected on the surface of the scalp. Indeed, at a given moment, a measurement of intensity of the electric in microvolt is made for each of the 128 electrodes. By associating all the electrodes, it is then possible to describe a spatial configuration of the electrical activity at time t, named topography or map. A topography is then the result of the spatial organization of the electrical intensity level of the 128 electrodes relative to each other (Figure 1). An ECD is generated during a specific time period, during which the group of neurons is synchronized. Thus, it seems possible to segment the ERP into a series of stable spatial configurations of electrical activity, separated by abrupt transitions (Michel el al. [2009]). The psycholinguist's job is then to associate each spatial configuration with the cognitive treatments.

The objective of this chapter is to classify the 128 electrodes for each time step. The data consists of 128 electrodes and 285 time steps and has the form of a electrode-by-time step matrix. For each time step T_i, we transform the electrodes-by-T_i matrix consisting of the electrodes and this time step into an electrodes-by-electrodes matrix of dimension 128×128. The considered network is then an undirected network without self loop built of 128 nodes for which each electrode corresponds to a node. In addition, each node is associated with a weight vector representing the absolute difference between the signal intensity of the electrode and that of the neighboring electrodes. The neighborhood is defined with respect to the positions of the electrodes on the cap. These are the near electrodes spatially. Since the electrical intensity may be positive or negative, a sign has been assigned for each edge between a pair of electrodes. This sign is positive if the valence of the intensity of the two nodes is the same (+ / + or - / -). It is negative if the valence is different for the two nodes (+/- or - / +). The considered network is then a binary network having weights attributed to the nodes. In order to classify these nodes, a stochastic block model has been developed. A variational approach is considered to estimate the parameters of the model as well as to classify these nodes. The goal is to cluster electrodes and then explore variations in averaged on the cluster intensity over time.

4.2 The Model

A node-weighted undirected network is represented by $G := ([n], X, A)$, where $[n]$ is the set of weighted nodes $\{1, ..., n\}$ for all $n \geq 1$, $A = (a_{iw})_{1 \leq i \leq n, 1 \leq w \leq d}$ is

the attributed weights to nodes matrix and X is the symmetric edge matrix of dimension $n \times n$ which encodes the observed interactions between nodes. We have, for all $i, j \in \{1, \ldots, n\}$,

$$X_{ij} = \begin{cases} 1 & \text{if the nodes } i \text{ and } j \text{ interact} \\ 0 & \text{otherwise.} \end{cases}$$

We assume that the number of groups in the network is fixed and chosen equal to Q ($Q \geq 1$). Let Z be a binary indicator matrix labeling node-to-community assignments. We have for all $i \in \{1, \ldots, n\}$ and $q \in \{1, \ldots, Q\}$,

$$Z_{iq} = \begin{cases} 1 & \text{if and only if node } i \text{ belongs to community } q \\ 0 & \text{otherwise.} \end{cases}$$

In the sequel, we assume that the edges X_{ij} and the attributed weights A_i are conditionally independent, given the community membership label.

4.3 Generation of Stochastic Blockmodel Data's

The stochastic blockmodel data's is supposed to be generated as follows

— The node-to-community assignments vectors Z_i, for $i \in \{1, \ldots, n\}$, are independent and sampled from a multinomial distribution as following

$$Z_i \sim \mathcal{M}(1, \alpha = (\alpha_1, \ldots, \alpha_Q)),$$

where $\alpha = (\alpha_1, \ldots, \alpha_Q)$ is the vector of class proportions of length Q such as

$$\sum_{q=1}^{Q} \alpha_q = 1.$$

— Each edge X_{ij} between the two nodes i and j is sampled from a Bernoulli distribution as follows

$$X_{ij} | Z_{iq} Z_{jl} = 1 \sim \mathcal{B}(\pi_{ql}),$$

where π is the matrix of connection probabilities between the clusters. Each entry π_{ql} represents the probability of existence of an edge between the q-labeled and l-labeled nodes, for all $q, l \in \{1, \ldots, Q\}$.

— The attributed weights to node vector A_i, for $i \in \{1, \ldots, n\}$, is sampled from multivariate Gaussian distribution as follows

$$A_i | Z_{iq} \sim \mathcal{N}(\mu_q, \Sigma_q),$$

where μ_q and Σ_q are respectively the mean vector of length d, and the covariance matrix of dimension $d \times d$ associated to the community q.

Note that the adjacency matrix X and the attributed weights to node matrix A are independent.

Let $\mu = (\mu_1, \ldots, \mu_Q)$ and $\Sigma = (\Sigma_1, \ldots, \Sigma_Q)$. In the following, We denote by θ the set of all the parameters to be estimated $\theta = (\alpha, \pi, \mu, \Sigma)$.

4.4 Variational Inference

Since the variable Z is latent, our model belongs to the class of incomplete data models. The log-likelihood of the incomplete data can be expressed as follows

$$\log \mathbb{P}_\theta(X, A) = \log \sum_z \mathbb{P}_\theta(X, A, Z), \tag{4.1}$$

where $\mathbb{P}_\theta(X, A, Z)$ is the likelihood of the complete data such as

$$\begin{aligned} \mathbb{P}_\theta(X, A, Z) &= \mathbb{P}_\pi(X|A, Z)\mathbb{P}_{\mu,\Sigma}(A|Z)\mathbb{P}_\alpha(Z) \\ &= \mathbb{P}_\pi(X|Z)\mathbb{P}_{\mu,\Sigma}(A|Z)\mathbb{P}_\alpha(Z), \end{aligned}$$

where

$$\begin{aligned} \mathbb{P}_\pi(X|Z) &= \prod_{i<j}^{n}\prod_{q,l}^{Q} \mathbb{P}_{\pi_{ql}}(X_{i,j}|Z_i, Z_j) \\ &= \prod_{i<j}^{n}\prod_{q,l}^{Q} \mathbb{P}_{\pi_{ql}}(X_{i,j})^{Z_{iq}Z_{jl}} \\ &= \prod_{i<j}^{n}\prod_{q,l}^{Q} \left(\pi_{ql}^{X_{ij}}(1-\pi_{ql})^{1-X_{ij}}\right)^{Z_{iq}Z_{jl}}. \end{aligned}$$

$$\begin{aligned} \mathbb{P}_{\mu,\Sigma}(A|Z) &= \prod_{i}^{n}\prod_{q}^{Q} \mathbb{P}_{\mu_q,\Sigma_q}(A_i|Z_i) \\ &= \prod_{i}^{n}\prod_{q}^{Q} \left(\frac{1}{(2\pi)^{d/2}|\Sigma_q|^{1/2}} e^{-\frac{1}{2}(A_i-\mu_q)^t\Sigma_q^{-1}(A_i-\mu_q)}\right)^{Z_{iq}}. \end{aligned}$$

and

$$\begin{aligned} \mathbb{P}_\alpha(Z) &= \prod_{i}^{n}\prod_{q}^{Q} \mathbb{P}_{\alpha_q}(Z_i) \\ &= \prod_{i}^{n}\prod_{q}^{Q} \alpha_q^{Z_{iq}}. \end{aligned}$$

The equation (4.1) is intractable since it requires a summation over all the possible values of Z. To tackle this issue, we have to use an iterative method. However, the expectation maximization (EM) algorithm requires the computation of $\mathbb{P}(Z|X)$ which is intractable because of the dependency of the edges X_{ij} as shown in the previous chapters. Hence, we use a variational approach to overcome the issue. We make use of the variational expectation maximization (VEM) algorithm defined in the previous chapters.

Variational Expectation Maximization algorithm

The log-likelihood can be decomposed as

$$\log \mathbb{P}_\theta(X, A) = \log \mathbb{P}_\theta(X, A, Z) - \log \mathbb{P}_\theta(Z|X, A).$$

By applying the conditional expectation

$$\begin{aligned} \mathbb{E}_{Z|X,A}[\log \mathbb{P}_\theta(X, A)] &= \mathbb{E}_{Z|X,A}[\log \mathbb{P}_\theta(X, A, Z)] - \mathbb{E}_{Z|X,A}[\log \mathbb{P}_\theta(Z|X, A)] \\ \log \mathbb{P}_\theta(X, A) &= \mathbb{E}_{Z|X,A}[\log \mathbb{P}_\theta(X, A, Z)] - \mathbb{E}_{Z|X,A}[\log \mathbb{P}_\theta(Z|X, A)]. \end{aligned} \tag{4.2}$$

Since the EM algorithm is intractable, we suggest to use a variational approach to tackle the issue. So, we replace $\mathbb{P}_\theta(Z|X, A)$ by an approximate distribution $R_{X,A}(Z)$.

By replacing $\mathbb{P}_\theta(Z|X, A)$ with $R_{X,A}(Z)$ in (4.2), we obtain

$$\begin{aligned} \log \mathbb{P}_\theta(X, A) &= \mathbb{E}_{R_{X,A}}[\log \mathbb{P}_\theta(X, A, Z)] - \mathbb{E}_{R_{X,A}}[\log \mathbb{P}_\theta(Z|X, A)] \\ &= \mathbb{E}_{R_{X,A}}[\log \mathbb{P}_\theta(X, A, Z)] + \mathbb{E}_{R_{X,A}}\left[\frac{\log R_{X,A}(Z)}{\log \mathbb{P}_\theta(Z|X,A)}\right] - \mathbb{E}_{R_{X,A}}[\log R_{X,A}(Z)] \\ &= \mathbb{E}_{R_{X,A}}[\log \mathbb{P}_\theta(X, A, Z)] + \mathrm{KL}(R_{X,A}(Z) \,\|\, \mathbb{P}_\theta(Z|X, A)) \\ &\quad - \mathbb{E}_{R_{X,A}}[\log R_{X,A}(Z)]. \end{aligned} \tag{4.3}$$

where $\mathrm{KL}(R_{X,A}(Z) \parallel \mathbb{P}_\theta(Z|X, A))$ is the Kullback-Leibler divergence between $\mathbb{P}_\theta(Z|X, A)$ and its approximate distribution $R_{X,A}(Z)$. It measures the closeness between them. So the aim here is to minimize $\mathrm{KL}(R_{X,A}(Z) \parallel \mathbb{P}_\theta(Z|X, A))$.

We define $J_\theta(R_{X,A}(Z))$ by

$$\begin{aligned} J_\theta(R_{X,A}(Z)) &= \log \mathbb{P}_\theta(X, A) - \mathrm{KL}(R_{X,A}(Z) \| \mathbb{P}_\theta(Z|X, A)) \\ &= \mathbb{E}_{R_{X,A}}[\log \mathbb{P}_\theta(X, A, Z)] - \mathbb{E}_{R_{X,A}}[\log R_{X,A}(Z)]. \end{aligned} \tag{4.4}$$

The second equality is is deduced from (4.3).

Since the Kullback-Leibler divergence KL is non-negative, then $J_\theta(R_{X,A}(Z))$ is a lower bound of $\log \mathbb{P}_\theta(X, A)$. Furthermore, since the log-likelihood $\log \mathbb{P}_\theta(X, A)$

does not depend on the distribution $R_{X,A}$, then maximizing the lower bound $J_\theta(R_{X,A}(Z))$ is equivalent to minimizing $\mathrm{KL}(R_{X,A}(Z) \parallel \mathbb{P}_\theta(Z|X,A))$.

By combining equations (4.1) and (4.4), we obtain

$$\begin{aligned} J_\theta(R_{X,A}(Z)) &= \mathbb{E}_{R_{X,A}}[\log \mathbb{P}_\theta(X,A,Z)] - \mathbb{E}_{R_{X,A}}[\log R_{X,A}(Z)] \\ &= H(R_X) + \sum_i \sum_q \mathbb{E}_{R_{X,A}}(Z_{iq}) \log \alpha_q + \sum_{i<j} \sum_{q,l} \mathbb{E}_{R_{X,A}}(Z_{iq}, Z_{jl})(X_{ij} \log \pi_{ql} \\ &\quad + (1 - X_{ij}) \log(1 - \pi_{ql})) + \sum_i \sum_q \mathbb{E}_{R_{X,A}}(Z_{iq})(-\log((2\pi)^{d/2}|\Sigma_q|^{1/2}) \\ &\quad - \frac{1}{2}(A_i - \mu_q)^t \Sigma_q^{-1}(A_i - \mu_q)), \end{aligned} \tag{4.5}$$

where $H(R_X) = -\sum_i \sum_q \mathbb{E}_{R_{X,A}}(Z_{iq}) \log \mathbb{E}_{R_{X,A}}(Z_{iq})$.

We assume that the latent variable $R_{X,A}(Z)$ can be factorized over the latent variable Z as follows

$$R_{X,A}(Z) = \prod_{i=1}^{n} R_{X,A,i}(Z_i) = \prod_{i=1}^{n} h(Z_i; \tau_i), \tag{4.6}$$

where $\{\tau_i \in [0,1]^Q, i = 1, \ldots, n\}$ are the variational parameters associated with $\{Z_i, i = 1, \ldots, n\}$ such as $\sum_q \tau_{iq} = 1, \ \forall i \in \{1, \ldots, n\}$ and h is the multinomial distribution with parameters τ_i.

By combining the two equations (4.5) and (4.6), we obtain

$$\begin{aligned} J_\theta(R_{X,A}(Z)) &= -\sum_i \sum_q \tau_{iq} \log \tau_{iq} + \sum_i \sum_q \tau_{iq} \log \alpha_q + \sum_{i<j} \sum_{q,l} \tau_{iq}\tau_{jl}(X_{ij} \log \pi_{ql} \\ &\quad + (1 - X_{ij}) \log(1 - \pi_{ql})) + \sum_i \sum_q \tau_{iq}(-\log((2\pi)^{d/2}|\Sigma_q|^{1/2}) \\ &\quad - \frac{1}{2}(A_i - \mu_q)^t \Sigma_q^{-1}(A_i - \mu_q)). \end{aligned} \tag{4.7}$$

The variational expectation maximization algorithm alternates between the following two steps :

— **Expectation step** : We fix θ, then we maximize the lower bound J with respect to τ. Under the condition $\sum_q \tau_{iq} = 1, \ \forall i \in \{1, \ldots, n\}$, we obtain $\hat{\tau}$ by the following fixed point relation

$$\hat{\tau}_{iq} \propto \alpha_q \left(\frac{1}{(2\pi)^{d/2}|\Sigma_q|^{1/2}} e^{-\frac{1}{2}(A_i - \mu_q)^t \Sigma_q^{-1}(A_i - \mu_q)} \right) \prod_j \prod_l \left(\pi_{ql}^{X_{ij}} (1 - \pi_{ql})^{m - X_{ij}} \right)^{\tau_{jl}}. \tag{4.8}$$

The estimation of τ is obtained from (4.8) by iterating a fixed point algorithm until convergence.

Proof. Following the same steps already done in chapter 2, we have

$$\begin{aligned} J_\theta(R_{X,A}(Z))+\lambda_i(\sum_q^Q \tau_{iq}-1) = &\sum_{i<j}\sum_{q,l}\tau_{iq}\tau_{jl}(X_{ij}\log\pi_{ql}+(1-X_{ij})\log(1-\pi_{ql})) \\ &+\sum_i\sum_q \tau_{iq}(-\log((2\pi)^{d/2}|\Sigma_q|^{1/2}) \\ &-\frac{1}{2}(A_i-\mu_q)^t\Sigma_q^{-1}(A_i-\mu_q))-\sum_i\sum_q\tau_{iq}\log\tau_{iq} \\ &+\sum_i\sum_q\tau_{iq}\log\alpha_q+\lambda_i(\sum_q\tau_{iq}-1). \end{aligned}$$

By deriving this equation with respect to τ_{iq} and by taking this quantity equal to zero, we obtain :

$$\begin{aligned} &\sum_l^Q\sum_{j=1,j\neq i}^n (X_{ij}\log\pi_{ql}+(1-X_{ij})\log(1-\pi_{ql}))\tau_{jl}+(-\log((2\pi)^{d/2}|\Sigma_q|^{1/2}) \\ &-\frac{1}{2}(A_i-\mu_q)^t\Sigma_q^{-1}(A_i-\mu_q))+\log(\alpha_q)-\log\tau_{iq}-1+\lambda_i=0. \end{aligned}$$

Then, deriving it with respect to λ_i and taking this quantity equal to zero, we obtain :

$$\sum_q^Q \tau_{iq}-1=0.$$

This leads to the following fixed point relation
$\forall i\in\{1,\dots,n\},\ \forall q\in\{1,\dots,Q\}$,

$$\begin{aligned} \hat{\tau}_{iq} &= e^{-1+\lambda_i}\alpha_q\left(\frac{1}{(2\pi)^{d/2}|\Sigma_q|^{1/2}}e^{-\frac{1}{2}(A_i-\mu_q)^t\Sigma_q^{-1}(A_i-\mu_q)}\right)\prod_j\prod_l\left(\pi_{ql}^{X_{ij}}(1-\pi_{ql})^{m-X_{ij}}\right)^{\hat{\tau}_{jl}} \\ &\propto \alpha_q\left(\frac{1}{(2\pi)^{d/2}|\Sigma_q|^{1/2}}e^{-\frac{1}{2}(A_i-\mu_q)^t\Sigma_q^{-1}(A_i-\mu_q)}\right)\prod_j\prod_l\left(\pi_{ql}^{X_{ij}}(1-\pi_{ql})^{m-X_{ij}}\right)^{\hat{\tau}_{jl}}. \end{aligned}$$

Recall that $\propto$ means "proportional to" and $e^{(-1+\lambda_i)}$ is the normalizing constant. □

— **Maximization step** : We are interested here in estimation θ so we fix τ, then we maximize the lower bound J with respect to each parameters.

— By maximizing J with respect to α and under the condition $\sum_q\alpha_q=1, \forall i\in\{1,\dots,n\}$, we obtain

$$\hat{\alpha}_q=\frac{1}{n}\sum_i\tau_{iq}.$$

The proof is given in chapter 1.

— By maximizing J with respect to π, we obtain

$$\hat{\pi}_{ql} = \frac{\sum_{i<j} \tau_{iq}\tau_{jl}X_{ij}}{\sum_{i<j} \tau_{iq}\tau_{jl}}.$$

Proof. The lower bound must be maximized with respect to π. We fix all the other parameters , then we maximize the lower bound (4.7) with respect to π_{ql}. By deriving (4.7) with respect to π_{ql} and by taking this quantity equal to zero, we obtain :

$$\sum_{i<j} \tau_{iq}\tau_{jl}\left(\frac{X_{ij}}{\pi_{ql}} - \frac{(1-X_{ij})}{(1-\pi_{ql})}\right) = 0.$$

This leads to the following estimate of π_{ql}

$$\hat{\pi}_{ql} = \frac{\sum_{i<j} \tau_{iq}\tau_{jl}X_{ij}}{\sum_{i<j} \tau_{iq}\tau_{jl}}.$$

□

— By maximizing J with respect to μ, we obtain

$$\hat{\mu}_q = \frac{\sum_i \tau_{iq}A_i}{\sum_i \tau_{iq}}.$$

Proof. The lower bound must be maximized here with respect to μ. We fix all the other parameters , then we maximize the lower bound (4.7) with respect to μ_q. By deriving (4.7) with respect to μ_q and by taking this quantity equal to zero, we obtain :

$$\sum_i \tau_{iq}(A_i - \mu_q)^t\Sigma^{-1} = 0.$$

This leads to the following estimate of μ_q

$$\hat{\mu}_q = \frac{\sum_i \tau_{iq}A_i}{\sum_i \tau_{iq}}.$$

□

— By maximizing J with respect to Σ, we obtain

$$\hat{\Sigma}_q = \frac{\sum_i \tau_{iq}(A_i - \hat{\mu}_q)(A_i - \hat{\mu}_q)^t}{\sum_i \tau_{iq}}.$$

First, we start by defining a theorem that we will use later in the proof.

Theorem 4.4.1. *(see Jordan [2010])*
The trace (denoted by tr) is invariant under cyclical permutations of matrix products

$$tr[ABC] = tr[CAB] = tr[BCA], \tag{4.9}$$

where A, B and C are are arbitrary matrices whose dimensions are compatible and are such that the product of the matrices ABC is a square matrix. Let V_i be a vector. Since the product x^tVx is then a scalar, then we have

$$x^tVx = tr[x^tVx]. \tag{4.10}$$

Let A and B be two arbitrary matrices whose dimensions are compatible, then

$$\frac{\partial}{\partial A} tr[BA] = B^t. \tag{4.11}$$

Let A be an arbitrary matrix. Then,

$$\frac{\partial}{\partial A} log|A| = A^{-t}. \tag{4.12}$$

The proof of the theorem is given in Jordan [2010].

Proof. Since we are interested here in calculating the estimation of Σ, we fix all the other parameters. According to (4.7), the lower bound estimate of the covariance matrix Σ is given by

$$l(\Sigma) = \sum_i \sum_q \tau_{iq}(-\frac{1}{2}\log|\Sigma_q| - \frac{1}{2}(A_i - \mu_q)^t\Sigma_q^{-1}(A_i - \mu_q)).$$

Using the fact that the determinant of the inverse of a matrix is the inverse of the determinant of the matrix, we obtain

$$l(\Sigma) = \sum_i \sum_q \tau_{iq}(\frac{1}{2}\log|\Sigma_q^{-1}| - \frac{1}{2}(A_i - \mu_q)^t\Sigma_q^{-1}(A_i - \mu_q)).$$

Now, since $(A_i - \mu_q)$ is a vector, then $(A_i - \mu_q)^t\Sigma_q^{-1}(A_i - \mu_q)$ is a scalar. Using 4.10, $l(\Sigma)$ can be expressed as

$$l(\Sigma) = \sum_i \sum_q \tau_{iq}(\frac{1}{2}\log|\Sigma_q^{-1}| - \frac{1}{2}\mathrm{tr}[(A_i - \mu_q)^t\Sigma_q^{-1}(A_i - \mu_q)]).$$

Then using 4.9, we obtain

$$l(\Sigma) = \sum_i \sum_q \tau_{iq}(\frac{1}{2}\log|\Sigma_q^{-1}| - \frac{1}{2}\mathrm{tr}[(A_i - \mu_q)(A_i - \mu_q)^t\Sigma_q^{-1}]).$$

Now, by deriving $l(\Sigma)$ with respect to Σ^{-1} and using 4.11 and 4.12, we obtain

$$\begin{aligned}\frac{\partial l}{\partial \Sigma^{-1}} &= \sum_i \sum_q \tau_{iq}\left(\frac{1}{2}\Sigma_q - \frac{1}{2}((A_i-\mu_q)(A_i-\mu_q)^t)^t\right) \\ &= \sum_i \sum_q \tau_{iq}\left(\frac{1}{2}\Sigma_q - \frac{1}{2}(A_i-\mu_q)(A_i-\mu_q)^t\right).\end{aligned}$$

Finally, setting to zero yields to

$$\hat{\Sigma}_q = \frac{\sum_i \tau_{iq}(A_i-\hat{\mu}_q)(A_i-\hat{\mu}_q)^t}{\sum_i \tau_{iq}}.$$

□

4.5 Selection Criterion

We propose to use the ICL criterion to estimate the most adequate number of clusters $\hat{Q}$ in the network. This criterion is already defined in the chapters 1 and 2.

The ICL can be expressed through

$$\begin{aligned}ICL(Q) &= \sum_{i<j}\sum_{q,l}\hat{\tau}_{iq}\hat{\tau}_{jl}(X_{ij}\log\hat{\pi}_{ql} + (1-X_{ij})\log(1-\hat{\pi}_{ql})) + \sum_i\sum_q \hat{\tau}_{iq}\log\hat{\alpha}_q \\ &+ \sum_i\sum_q \hat{\tau}_{iq}(-\log((2\pi)^{\frac{d}{2}}|\hat{\Sigma}_q|^{\frac{1}{2}}) - \frac{1}{2}(A_i-\hat{\mu}_q)^t\hat{\Sigma}_q^{-1}(A_i-\hat{\mu}_q)) \\ &- \frac{1}{2}\left(\frac{Q(Q+1)}{2}\log\frac{n(n-1)}{2} + (Q-1)\log n + Qd\log n\right. \\ &\left. + Q\frac{d(d+1)}{2}\log n\right).\end{aligned}$$

Our algorithm is run for different values of Q, then $\hat{Q}$ is chosen such that the ICL is maximized.

4.6 Application to EEG Data

Using the fitting of the SBM model, the analysis revealed a set of 4 clusters of electrodes. Figure 4.1 shows the spatial distribution on the scalp surface of each cluster.

The objective is to explore the evolution of the averaged intensity of clusters over time. More precisely, we seek to reveal the temporal periods of change of cerebral localization of ECDs. As explained above, the cognitive process is based on

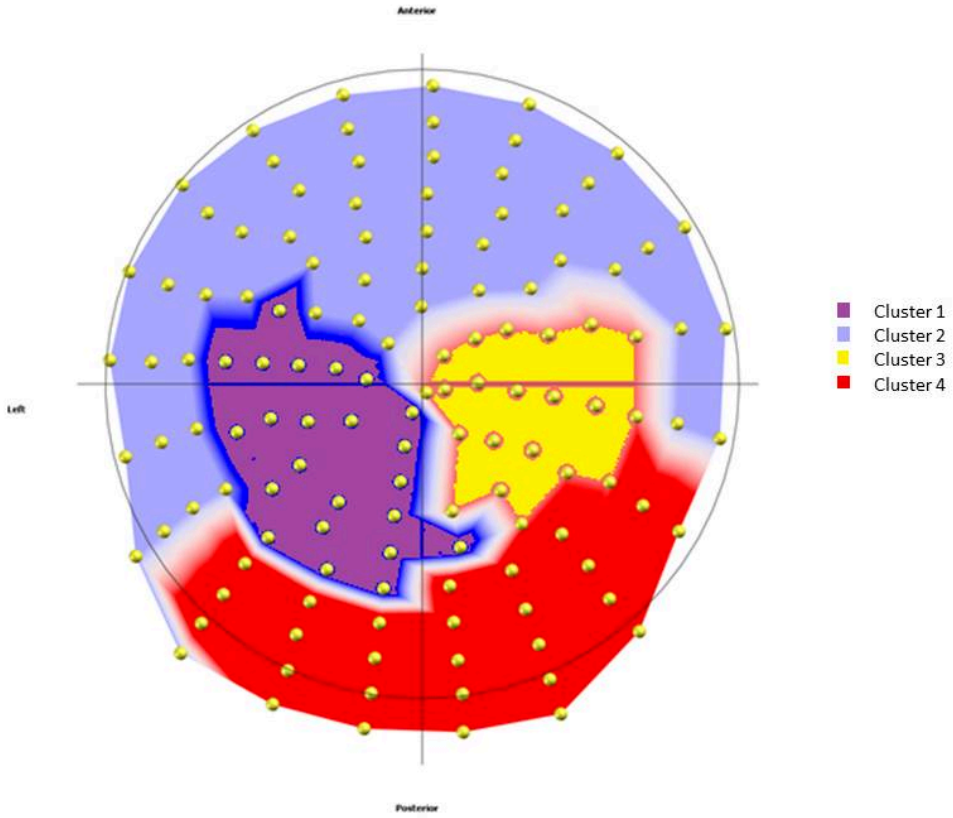

FIGURE 4.1 – EEG data : Spatial distribution of the four clusters.

periods during which one (or more) group(s) of specific neurons is (are) synchronized. During these periods, stable topographies are observed on the scalp (Michel el al. [2009]). In addition, abrupt changes occur between these periods. It should therefore be possible to specify these change periods from the clustering analysis. The periods of change are characterized by changes in the electrical intensity measured for specific electrodes. A measure of the overall intensity of each cluster and its evolution over time should then allow us to highlight these periods of rupture.

In order to be able to establish if this approach makes it possible to highlight the periods of change in brain activity, we can compare the results reported above with those for a component of the brain activity involved in the naming of images : the P100. It is an occipital component, appearing during a time window beginning at 75 ms and ending at 150 ms after the presentation of the image. This is the topography shown in Figure B.1. It has an occipital location and is associated with visual processing.

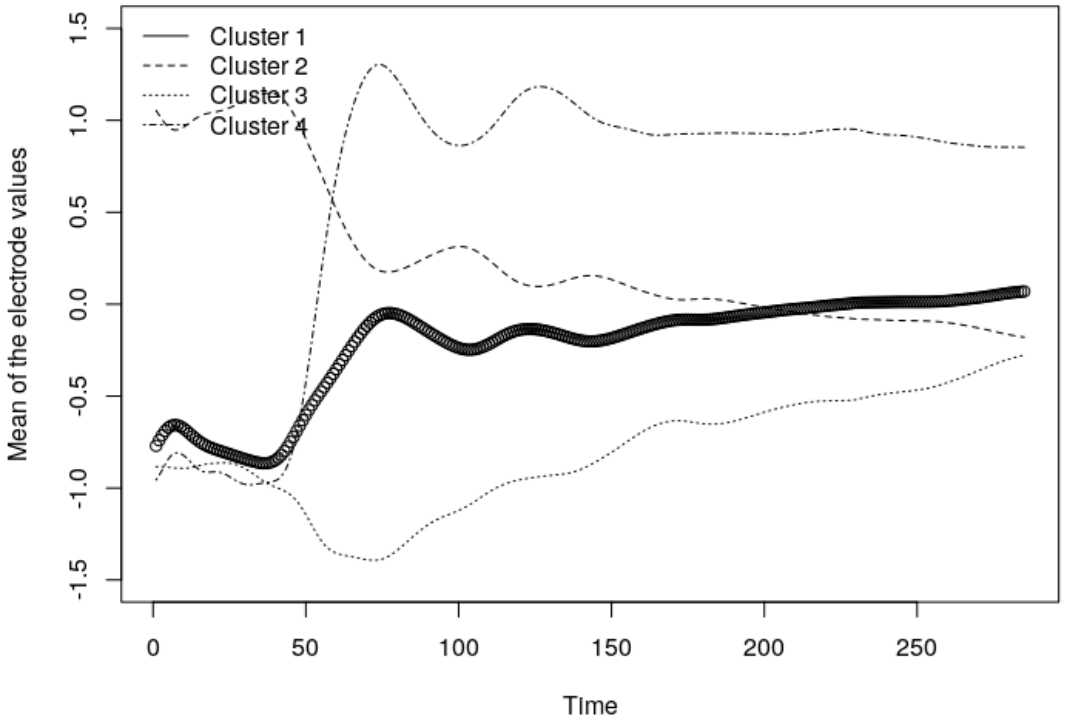

FIGURE 4.2 – Evolution of the mean of the electrodes values in different clusters.

Figure 4.2 shows averaged intensity inflections of clusters. Even though additional analyzes will have to be performed, it seems that the clustering analysis described here makes it possible to specify the periods of change of the electrical activity of the brain from the analysis of the time points of change of the average intensity of the electrodes of each cluster. These first analyzes confirm that this new modelling approach to `EEG` data treatment is very promising.

Chapitre 5

Conclusion et Perspectives

Ce travail porte sur la classification des réseaux en utilisant des modèles à blocs stochastiques. Nous développons des méthodes d'inférence basé sur des algorithmes variationnels pour estimer les paramètres du modèle proposé ainsi que pour classifier les noeuds du réseau considéré.

Dans le premier chapitre, nous avons développé une introduction générale du travail en introduisant des méthodes de classification classiques puis nous avons défini les modèles à blocs stochastiques pour classifier les réseaux binaires. Nous avons ensuite développé une méthode d'inférence basée sur l'algorithme espérance maximisation variationnel (`VEM`) afin d'estimer les paramètres du modèle et de classifier les sommets du réseau. En effet, puisque le log de la vraisemblance des données incomplètes $\log \mathbb{P}_{\theta(X)} = \log \sum_Z \mathbb{P}_\theta(X, Z)$ est intraitable sauf pour les réseau ayant un petit nombre de noeuds n, nous avons utilisé l'algorithme espérance maximisation (`EM`) pour résoudre ce problème. D'autre part, puisque les arêtes joignant les sommets du réseau ne sont pas indépendantes, le calcul de la distribution de la variable latente sachant la variable observée $\mathbb{P}(Z|X)$ est impossible et de ce fait l'étape espérance de l'algorithme `EM` qui nécessite le calcul de $\mathbb{P}(Z|X)$ est intraitable. Pour cela, nous avons développé l'algorithme espérance maximisation variationnel. Cet algorithme alterne deux étapes. La première consiste à estimer la variable latente alors que la deuxième consiste à estimer les paramètres du modèle proposé.

Dans le chapitre 2, nous avons défini un modèle à blocs stochastiques binomial pour classifier les réseaux de co-citations dans un contexte de fouille de textes. Ces réseaux sont pondérés. Chaque arête joignant une paire de terme est pondérée en fonction du nombre de documents citant simultanément cette paire de termes. Nous avons développé un algorithme `VEM` pour estimer les paramètres du modèle ainsi que pour classifier les noeuds du réseau. Puis, nous avons introduit un critère de sélection du modèle optimal basé sur le critère `ICL` (en anglais integrated classification likelihood). Ceci nous permet de choisir le nombre optimal de clusters

qui correspond au modèle. Nous avons finalement comparé le modèle proposé avec le modèle à blocs stochastiques avec des arêtes distribuées selon une loi de Poisson. Les résultats montrent que la méthode proposée donne de meilleurs résultats que l'autre méthode et que le temps de convergence de calcul de cet algorithme est satisfaisant. En outre, cette méthode est aisée à implémenter en utilisant le logiciel R.

Dans le chapitre 3, nous avons développé la méthode espérance maximisation variationnelle bayésienne (`VBEM`) pour estimer les paramètres dans un modèle à blocs stochastiques binomial. Cette question est motivée par les réseaux de co-citations dans un contexte de fouille de textes comme l'indique le chapitre 2. Ensuite, nous avons développé un critère `ILvb` (en anglais integrated likelihood variational Bayes) pour sélectionner le nombre optimal de classes. Nous avons enfin comparé la méthode proposée avec le `VEM` en appliquant ces deux approches sur un ensemble de données réelles puis sur un corpus d'entretiens de mineurs migrants, de la région subsaharienne à la côte européenne méditerranéenne. Nous avons appliqué la méthode proposée pour classifier les 25 termes les plus pertinents à partir d'une liste de termes utilisés dans les entretiens avec les mineurs migrants et classé par fréquence dans l'ensemble du corpus. Nous avons comparé la méthode proposée avec le `VEM`.

Dans le chapitre 4, nous avons développé un modèle à blocs stochastiques afin de classifier un réseau binaire ayant des vecteurs de poids attribués au noeuds. Ce modèle prend deux matrices en tant que données d'entrée, l'une est la matrice d'adjacence du graph et l'autre est la matrice de pondération associée aux noeuds. Cette question est motivée par la classification des différents traitement cognitifs réalisé par le cerveau lors de la préparation à l'écriture à partir de l'activité électrique produite par les neurones du cerveau, traitée et enregistrée par l'électroencéphalogramme. Le réseau considéré possède 128 noeuds pondérés. Chaque noeud correspond à une électrode associée à un vecteur de poids représentant la différence absolue entre l'intensité du signal de cette électrode et celle de ses voisins. De plus, une arête joignant une paire d'électrodes est présente si ces deux électrodes ont le même signe d'intensité électrique (+ / + ou - / -). Nous avons développé la méthode `VEM` pour estimer les paramètres du modèle ainsi que pour classifier les noeuds du réseau. Nous avons ensuite introduit un critère `ICL` pour estimer le nombre optimal de clusters dans le réseau.

Dans un travail ultérieur, nous souhaitons généraliser ce travail pour traiter le cas des réseaux multiplex pondérés où une ou plusieurs arêtes pondérées peuvent exister entre une paire de noeuds. Cette question est motivée par l'existence de plusieurs relations pondérées de différents types entre les paires de noeuds. Nous souhaitons ensuite appliquer ce travail sur un ensemble de données de pollution de la rivière "Litani" au Liban afin de classifier des paramètres physico-chimiques

obtenus sur chaque station de cette rivière entre la période 2009 et 2016.

D'autre part, nous allons reprendre le corpus d'entretiens des mineurs migrants de la région subsaharienne à la côte européenne méditerranéenne détaillé dans le chapitre 3, section 3.7, pour classifier un nombre plus important de termes utilisés dans les entretiens. De plus, nous souhaitons reprendre ce jeu de donnés en considérant cette fois les relations entre les mineurs migrants (en tenant compte de mots qu'ils utilisent en commun dans les entretiens) au lieu des mots cocités. Et ceci en considérant un modèle de biclustering (`LBM`) (Brault and Mariadassou [2015]).

Bibliographie

Airoldi, E., Blei, D., Fienberg, S., and Xing, E. (2008). Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, **9**, 1981–2014.

Anderberg, M. R. (1973). Cluster analysis for applications. *Office of the Assistant for Study Support Kirtland AFB N MEX*.

Anderson, C. J., Wasserman, S., and Faust, K. (1992). Building stochastic blockmodels. *Social Networks*, **14**, 137–161.

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational bayes. In : *Laskey K, Prade H (eds) Uncertainty in Artificial Intelligence : proceedings of the fifth conference*, Morgan Kaufmann, 21–30.

Ball, B., Karrer, B., and Newman, M. E. (2011). Efficient and principled method for detecting communities in networks. *Physical Review E*, **84**, 036–103.

Barbillon, P., Donnet, S., Lazega, E., and Bar-Hen, A. (2017). Stochastic block models for multiplex networks : an application to a multilevel network of researchers. *Journal of the Royal Statistical Society : Series A (Statistics in Society)*, **180**, 295–314.

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

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference : A review for statisticians. *Journal of the American Statistical Association*, **112**, 859–877.

Brault, V., and Mariadassou, M. (2015). Co-clustering through latent bloc model : A review. *Journal de la Société Française de Statistoque*, **156**, 120–139.

Breiger, R. L. (1974). The duality of persons and groups. *Social Forces*, **53**, 181–190.

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

Davies, A., Gardner, B. B., and Gardner, M. R. (1941). Deep South, *The University of Chicago Press*.

Demmel, J. (1999). Graph Partitioning, Part 2. https://people.eecs.berkeley.edu/demmel/cs267/lecture20/lecture20.html.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, p. 1–38.

Erdös, P. and Rényi, A. (1960). On the evolution of random graphs. *Periodica Mathematica Hungarica*, **5**, 17–60.

Feinerer, I., Hornik, K. and Meyer, D. (2008). Text mining infrastructure in R. *Journal of Statistical Software*, **25**, 1–54.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, **486**, 75–174.

Freeman, L. C. (1993). On the sociological concept of "group" : a empirical test of two models. *American Journal of Sociology*, **98**, 152–166.

Freeman, L. C. (1994). Finding groups with a simple genetic algorithm. *Journal of Mathematical Sociology*, **17**, 227–241.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., and Airoldi, E. M. (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning*, **2**, 129–233.

Holland, P. W., Laskey, K. B., and Leinhardt S. (1983). Stochastic blockmodels : First steps. *Social Networks*, **5**, 109–137.

Hubert, L., and Arabie, P. (1985). Comparing partitions. *Journal of classification*, **1**, 193–218.

Jaakkola. T. S. (2000). Tutorial on variational approximation methods. *Advanced mean field methods : theory and practice*, 129–159.

Jaakkola, T. S., and Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, **10**, 25–37.

Jeffreys, H. (1946). An invariant form for the prior probability in estimations problems. *Proceedings of the Royal Society of London. Series A*, **186**, pp 453–46.

Jernite, Y., Latouche, P., Bouveyron, C., Rivera, P., Jegou, L., and Lamassé, S. (2014). The random subgraph model for the analysis of an acclesiastical network in merovingian gaul. *Annals of Applied Statistics*, **8**, 55–74.

Jordan, M. (2010). Bayesian Modeling and Inference. *Springer*. https://people.eecs.berkeley.edu/ jordan/courses/260-spring10/other-readings/ chapter13.pdf

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, **37**, 183–233.

Ke, W., Borner, K. and Viswanath, L. (2004). Major information visualization authors, papers and topics in the acm library. *IEEE symposium on information visualization*.

Karrer, B. and Newman, M.E.J. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, **83**, 016–107.

Latouche, P., Birmelé, E., and Ambroise, C. (2011). Overlapping stochastic block models with application to the French political blogosphere. *Annals of Applied Statistics*, **5**, 309–336.

Latouche, P., Birmelé, E., and Ambroise, C. (2012). Variational Bayesian inference and complexity control for stochastic block models. *Statistical Modelling*, **12**, 93–115.

Lei, J. (2016). A goodness-of-fit test for stochastic block models. *The Annals of Statistics*, **1**, 401–424.

Lewis, D. (1997). Reuters-21578 Text Categorization Collection Distribution 1.0. http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

Linton C. (2003). Finding Social Groups : A Meta-Analysis of the Southern Women Data, In Ronald Breiger, Kathleen Carley and Philippa Pattison, eds. Dynamic Social Network Modeling and Analysis. *The National Academies Press*.

Louis, P.-Y., Robin, N. (2016). Une mobilité d'une extraordinaire singularité : les mineurs de l'Afrique subsaharienne aux rives sud de la Méditerranéee. *Comprendre les migrations internationales : retour sur 30 ans de recherches, ISBN Presses Universitaires* Francois-Rabelais de Tours.

Luck SA.(2005). An introduction to the event-related potential technique. *The MIT Press*, 7–21.

MacQueen, J.(1968). Some methods for classification and analysis of multivariate observation. *In proceeding of the fifth Berkley symposium on mathematical statistics and probability*, **1**, 281–297.

Mariadassou, M., Robin, S., and Vacher, C. (2010). Uncovering latent structure in valued graphs : a variational approach. *Annals of Applied Statistics*, **4**, 715–742.

Matias, C., and Miele, V. (2017). Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, **79**, 1119–1141.

McLachlan, G., and Krishnan, T. (2007). The EM algorithm and extensions. *John Wiley and Sons*, **382**.

Michel C., Koenig T., Brandeis D. (2009). Electrical neuroimaging in the time domain. *Michel CM, Koenig T, Brandeis D, Gianotti LRR and Wackermann J, Vol. Electrical Neuroimaging*, Cambridge, 111–143.

Nowicki, K., and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association*, **96**, 1077–1087.

Osbourn, G. C., and Martinez, R. F. (1995). Empirically defined regions of influence for clustering analyses. *Pattern Recognition*, **28**, 1793–1806.

Perret, C., and Laganaro, M. (2012). Comparison of electrophysiological correlates of writing and speaking : a topographic ERP analysis. *Brain topography*, **25**, 64-72.

Perret, C., and Olive, T. (2019). Writing Words : a Brief Introduction. *Spelling and Writing Words*, 1–15.

Porter, M. A., Onnela, J. P., and Mucha, P. J. (2009). Communities in networks. *Notices of the American Mathematical Society*, **56**, 1082–1097.

Rokach, Lior, and Oded Maimon. (2005). "Clustering methods." *Data mining and knowledge discovery handbook*, 321–352.

Snijders, T. A., and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, **14**, 75–100.

Svensén M., Bishop, C. (2004). Robust bayesian mixture modelling. *Neurocomputing*, **64**, 235–252.

Thomas, A.C. and Blitzstein, J.K. (2011). Valued ties tell fewer lies : Why not to dichotomize network edges with thresholds. arXiv :1101–0788.

Xu, K., and Hero III, A. (2013). Dynamic stochastic blockmodels : Statistical models for time-evolving networks. *Social Computing, Behavioral-Cultural Modeling and Prediction*. Springer, 201–210.

Yang, T., Chi, Y., Zhu, S., Gong, Y., and Jin, R. (2011). Detecting communities and their evolutions in dynamic social networks : a bayesian approach. *Machine learning*, **82**, 157–189.

Zhao, Y. (2012). R and data mining : Examples and case studies. *Academic Press*

Zreik, R., Latouche, P., and Bouveyron, C. (2017). The dynamic random subgraph model for the clustering of evolving networks. *Computational Statistics*, **32**, 501–533.