Statistical analysis of networks and applications in Social Sciences

Rawya Zreik

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

THÈSE

présentée par
ZREIK Rawya

pour obtenir le grade de
DOCTEUR EN MATHÉMATIQUES APPLIQUÉES

Analyse statistique des réseaux et applications aux Sciences Humaines

réalisée sous la direction de
Charles Bouveyron & Pierre Latouche

Soutenue publiquement le 30 novembre 2016
devant le jury composé de

M.	**Biernacki**	**Christophe**	**Université Lille 1 & Inria**	(Rapporteur)
M.	**Chiquet**	**Julien**	**AgroParisTech & Inra**	(Rapporteur)
Mme.	**Cottrell**	**Marie**	**Université Paris 1**	(Examinateur)
Mme.	**Matias**	**Catherine**	**CNRS & Université Paris 6**	(Examinateur)
M.	**Come**	**Etienne**	**IFSTTAR**	(Examinateur)
M.	**Velcin**	**Julien**	**Université Lyon 2**	(Examinateur)
M.	**Lamassé**	**Stéphane**	**Université Paris 1**	(Examinateur)
M.	**Bouveyron**	**Charles**	**Université Paris 5**	(Directeur)
M.	**Latouche**	**Pierre**	**Université Paris 1**	(Encadrant)

Thèse préparée au sein du laboratoire **SAMM**

Acknowledgments

I would like to take this opportunity to thank the people with whom I have worked for the last three years, who have made the development of this thesis possible.

Charles and Pierre, thanks a lot for this chance that you gave me to be in your team, for your support and thesis supervision.

I would especially like to thank my husband and family. Thank you for being with me always, you have given me confidence and helped me to advance.

Lastly, a big thanks to all members of the SAMM laboratory as well as the MAP5 laboratory, especially Marie Cottrell and Jean-Marc Bardet.

Thank you all for everything.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Summary of the notations used in this thesis

Notations	Description
$\mathcal{G}$	Presents a graph .
X	Adjacency matrix, presenting either binary or categorical or textual edges between nodes.
Z	Binary matrix, pinpointing each node and its cluster.
N	Number of vertices in the network or total number of words.
L	Number of links in the graph.
Q	Number of latent clusters.
K	Number of latent topics.
S	Number of subgraphs.
C	Number of edge types.
T	Number of time points.
D	Number of documents.
V	Representing the total number of vocabulary in the full set of documents.
W	Sequence of whole words on D document and W_n^d is the nth word in the document d.
Π	Describes the probabilities of connection between clusters, and Π_{ql}^c is the probability of having an edge of type c between vertices of clusters q and l.
α	Shows the proportion of clusters.

CHAPTER 1

INTRODUCTION

Over the last two decades, network structure analysis has experienced rapid growth with its construction and its intervention in many fields, such as: communication networks, financial transaction networks, gene regulatory networks, disease transmission networks, mobile telephone networks. Social networks are now commonly used to represent the interactions between groups of people; for instance, ourselves, our professional colleagues, our friends and family, are often part of online networks, such as Facebook, Twitter, email.

Since Moreno's original work on the network in 1934, research on the development of network structure has increased, and is still much debated, and over time, there has been an unprecedented rise in the amount of network data available.

1.1 Framework and contributions of the thesis

In a network, many factors can exert influence or make analyses easier to understand. Among these, we find two important ones: the time factor, and the network context. The former involves the evolution of connections between nodes over time. The network context can then be characterized by different types of information such as text messages (email, tweets, Facebook, posts, etc.) exchanged between nodes, categorical information on the nodes (age, gender, hobbies, status, etc.), interaction frequencies (e.g., number of emails sent or comments posted), and so on. Taking into consideration these factors can lead to the capture of increasingly complex and hidden information from the data.

The aim of this thesis is to define new models for graphs which take into consideration the two factors mentioned above, in order to develop the analysis of network structure and allow extraction of the hidden information from the data. These models aim at clustering the vertices of a network depending on their

connection profiles and network structures, which are either static or dynamically evolving. The starting point of this work is the stochastic block model, or SBM. This is a mixture model for graphs which was originally developed in social sciences. It assumes that the vertices of a network are spread over different classes, so that the probability of an edge between two vertices only depends on the classes they belong to. Despite the good performance of the clustering methods associated with this model on static networks, they are known to underperform when trying to take into consideration the network context for static networks, and also dealing with dynamic networks.

In this thesis, we intend to underline the problems which arise when using the SBM model. Therefore, we will offer solutions regarding the network structure analysis for different situations, either static or dynamic, and in various contexts. Thus, we undertake, on the one hand, defining a new model based on SBM to deepen the understanding of the network structure of dynamic networks. On the other hand, we look to understand the topology of the network, using both the connectivity between nodes and the context. To this end, we have three principal contributions, which are as follows.

In the first contribution, we propose a new random graph model, where we focus on modeling dynamic networks when taking into consideration time and the type of edges, which can be either categorical or binary. In this context, we try to discover the hidden characteristics and properties that explain a network over time, where a decomposition of the networks into subgraph is given. Once we have observed a categorical edge structure, we essentially treat the evolution of connections between nodes into subgraphs over time for the dynamic network. The subgraphs are assumed to be made of latent clusters which have to be inferred from the data in practice. The vertices are then connected with a probability depending only on the subgraphs whereas the edge type is assumed to be sampled conditionaly on the latent groups. Figure 1.1 illustrates an example of a dynamic network of a maritime network between 50 ports over the 4 years before and after the collapse of the Union of Soviet Socialist Republics (USSR). This example shows a real application of our methodology where the known subgraphs correspond to political systems in countries. Application of the proposed model allows us to:

1. Discover the network structure over time, capturing the evolution of connections between nodes over time from both the network's graph and the categories of edges.

2. Detect communities and their behavior from binary or categorical interactions over time, depending on each of the subgraphs.

3. Find the probability of connections between communities of nodes present in the network.

4. Predict the edges from new nodes, based on types of intersections and their subgraphs.

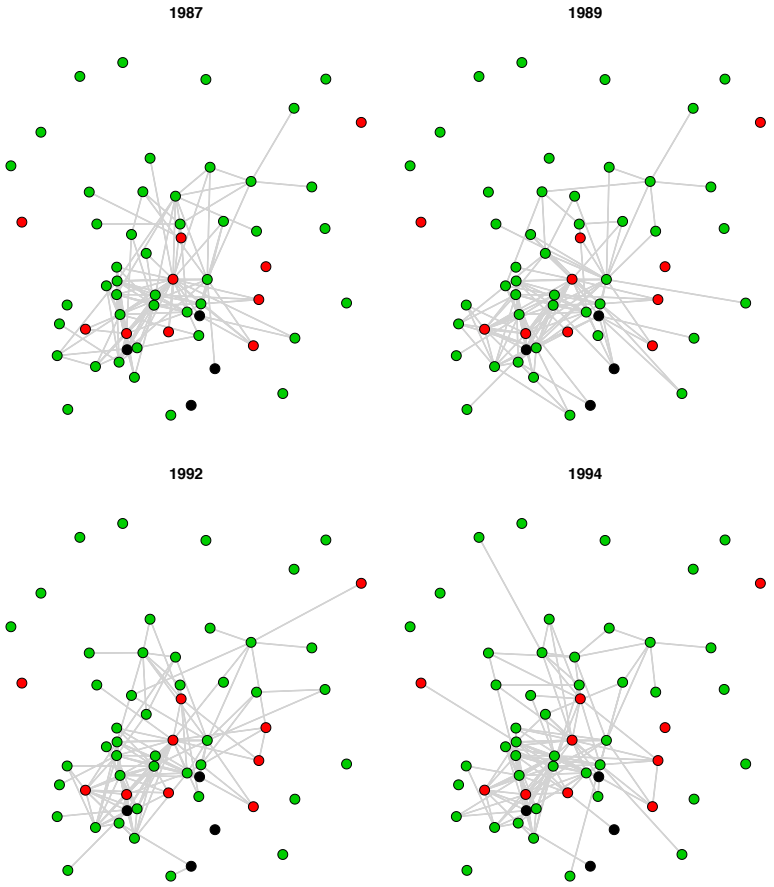


Figure 1.1: Maritime network between 50 ports over the 4 years before and after the collapse of the USSR. The known subgraphs correspond to political systems in countries, indicated using colors.

In our second contribution, we propose a new model to address the problem of finding topically meaningful clusters, by leveraging both links between individuals and text content shared between them in the static network. To this end, we are primarily concerned with accomplishing two tasks. The first one is to examine texts from social networks. The second one is to improve the understanding of the node's identities, by incorporating network context into network analysis algorithms, and by analyzing both the context and the graph. In this model, we can cluster or categorize identities of nodes according to "metadata" attached to these nodes. For example, this includes text messages (from tweets, Facebook, posts, etc.), interaction frequencies (emails), and more. Figure 1.2 illustrates a real static network to which we can apply our methodology. This network describes electronic communications between 148 employees at the famous Enron[1] company. Application of this second new model allows us to:

1. Discover network structure from both the network graph and context. For example, finding social network communities characterized both by link patterns and textual discourse.

2. Detect communities from textual interaction, such as emails and comments, etc.

3. Predict the edges to/from new nodes based only on textual data. For example: emails, newly written academic papers, etc.

4. Treat a graph as information flow between individuals and find sets of topics which summarize the subjects of network by studying this information.

Finally, the last contribution is to validate the performance of our new models by applying each one to several real data sets. To this end, on the one hand, we applied the dRSM model to two real world networks: the first one, describes the electronic communications between employees in the Enron company. The second one describes the maritime flows in the word. On the other hand, we applied STBM to the Enron email and the Nips co-authorship networks

1.2 Organization of the thesis

The *first chapter* of the thesis describes the current state-of-the-art in this domain, from which our new results come. Some general notation is given, and the most well-known clustering methods for network analysis are reviewed. These methods are mainly focused on statistical models, along with other models for inference, such as the expectation maximization (EM) algorithm and the variational EM algorithm. We also focus on some model selection criteria to

[1] `https://www.cs.cmu.edu/~./enron/`

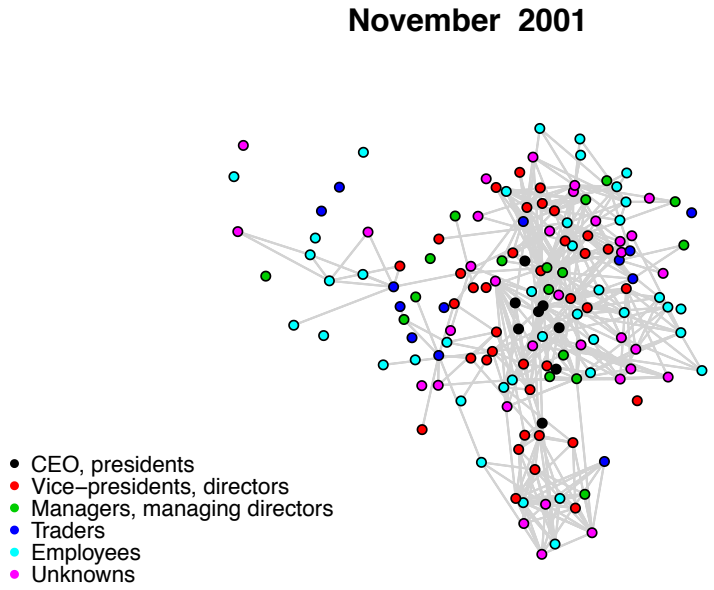


Figure 1.2: Static network describing electronic communications (edges) between 148 Enron employees (nodes), where each node color corresponds to the status of employees in Enron in November 2001.

estimate the number of classes from the data. Lastly, we focus on tools that we can integrate into the network, such as state space models (SSM) and latent dirichlet allocation (LDA). Several applications are presented in order to explain and clarify the model.

The *second chapter* presents our new model to analyze dynamic networks, which we call the dynamic random subgraph model (dRSM). This model is an extension of the random subgraph model (RSM) which was recently defined in order to deal with dynamic networks, using a state space model to characterize cluster proportions. A variational expectation maximization, or VEM, algorithm is proposed to approximate the posterior distribution over the model parameters and latent variables, which leads to a new state space model.

The *third chapter* defines the stochastic topic block model (STBM), which is a new model built to analyze networks with textual edges. STBMs aim to discover meaningful clusters of vertices that are coherent from both the network interaction and text content points of view. A classification variational expectation maximization (C-VEM) algorithm is proposed to perform inference.

Lastly, in the *fourth chapter*, we apply the two new models to real data sets. First, we show the capacity of dRSM to capture network dynamics, in order to uncover the evolution of clusters over time, in two different data sets. The first data presents a social network describing the electronic communications between employees. The second one is a geographical network which describes the maritime flows in the word. Secondly, we apply STBM to two different social networks: the Enron email and the Nips co-authorship networks, showing the ability of our methodology to detect communities, according to links and texts between nodes.

1.2.1 The main Publications

The main results of this thesis have been published in 3 articles (3 published), as well as one book chapter, which are:

- (Zreik et al., 2015): R. Zreik, P. Latouche, and C. Bouveyron. Classification automatique de réseaux dynamiques avec sous-graphes: étude du scandale enron. Journal de la Société Française de Statistique, 156(3):166–191, 2015.

- (Latouche et al., 2015): P. Latouche, R. Zreik, and C. Bouveyron. Cluster identification in maritime flows with stochastic methods. Maritime Networks: Spatial Structures and Time Dynamics, Routledge, 2015.

- (Zreik et al., 2016): R. Zreik, P. Latouche, and C. Bouveyron. The dynamic random subgraph model for the clustering of evolving networks. Computational Statistics, in press, 2016.

- (Bouveyron et al., 2016): C. Bouveyron, P. Latouche and R. Zreik. The stochastic topic block model for the clustering of vertices in networks with textual

edges. Statistics and Computing, pages DOI, 2016.

CHAPTER 2

STATE OF THE ART

Contents

This chapter is the introduction to several state of the art methods dealt with in this thesis. Section 2.1 introduces the general concept wherein the theory and general notations are provided. Section 2.3 defines some recent statistical methods for the modeling of static networks with an emphasis on the clustering of vertices and the estimation of model parameters. Then, in Section 2.4, are included models capable of handling dynamic networks. Section 2.5 defines tools that we later integrate to network models, to improve its analysis. In particular, we present tools to find the latent topics among documents, and tools to capture the evolution of connections between observations over time. Lastly, Section 2.6 illustrates, on the one hand, some variational techniques which lie at the core of the main inference strategies for networks developed in this thesis. On the other hand, we introduce some criteria to estimate the number of classes in networks.

2.1 Basics of the graph theory

The development of scientific studies has encouraged the rising need for network analysis in all fields, for instance in biology, history, geography, social media, etc. Since 1730, when Leonhard Euler published his paper on the problem of the Seven Bridges of Königsberg (Biggs et al., 1976), the graph theory has received strong attention from mathematical researchers, computer scientists, physicists, and sociologists. This research field allows the modeling of complex systems by characterizing pairwise interactions between objects of interest.

A graph is simply a collection of connected objects. We refer to these objects as "vertices" or "nodes". A node might be an individual, a computer, a site or even some geographical location. The connections between vertices are then defined by "edges" also called "links" (see Figure 2.1). In terms of vocabulary, the terms "network" and "graph" can be used as synonyms. In practice, the term "graph" is mainly used when characterizing the mathematical structure while "network" usually refers to the graph and all information available on it.

2.1.1 Types of graphs

Depending on the types of pairwise information provided in a data set, various types of graphs can be considered for modeling. The types of graph depend essentially on the types of edges and the presence of data on the edges.

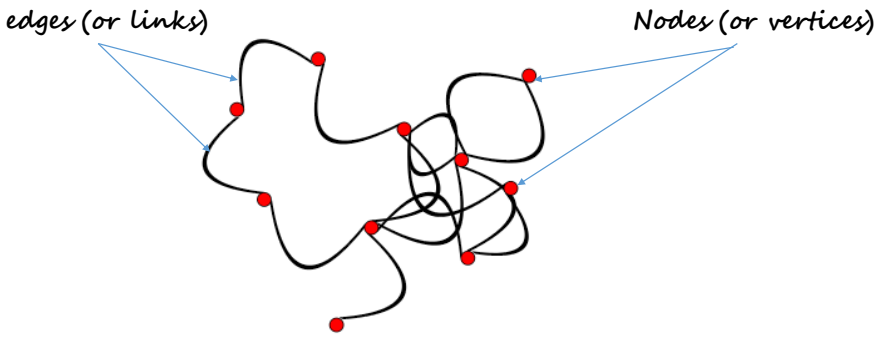


Figure 2.1: An undirected network with 10 nodes (or vertices) and 15 edges (or links).

Undirected / directed If the connections between vertices are not oriented, i.e. if vertex i is linked to j, then j is linked to i, the graph is undirected. Conversely, if relationships are oriented, then the graph is directed. For instance, in friendships networks, characterizing the recorded friendship links of students in a school, it is a common practice to find students naming others as friends with no reciprocity. Examples of undirected networks can be found for instance in biology with the use of protein networks to describe the binding of proteins. Two proteins are linked if their are known to bind in the cells. Figure 2.2 presents an example of both directed and undirected networks.

Static / dynamic If the connections between vertices are fixed over time, the data can be modeled as a static graph with fixed nodes and edges. An example of a static network is provided in Figure 2.3. However, as we shall see in Chapter 3, most networks used in real applications are dynamic with nodes and/or edges evolving over time. Nodes can appear or vanish. For instance, in a network characterizing the hyperlinks between websites, it is common to have new websites being created or closed. In Figures 2.4 and 2.5 the vertex set is fixed. However, the presence or absence of links between vertices change over time.

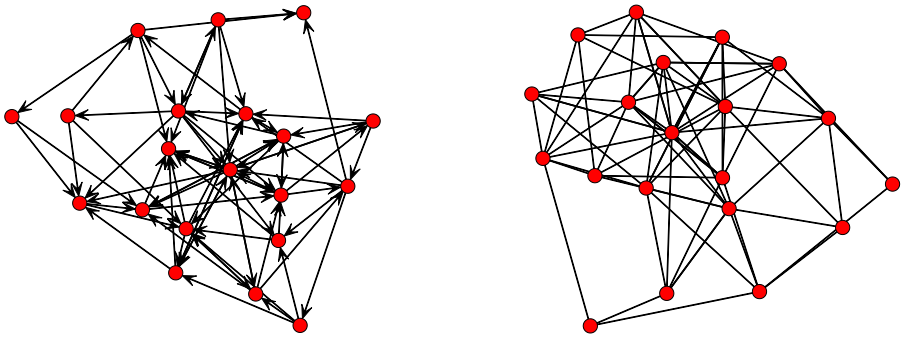


Figure 2.2: An example for both directed and undirected networks with 20 nodes.

Type of edges When modeling a real data set as an network, the starting point is usually to describe the presence or absence of pairwise relationships between vertices. In that case, edges can be characterized as binary variables with 1 indicating the existence of an edge, and 0 otherwise. Then, information on the edges can also be available. Indeed, edges can be attached to values in a given set. This is the case for instance when considering similarity measures or distances between species in a network. Graphs with binary edges are called binary graphs while they are called valued or weighted graphs if they are made of valued edges. If the graph allows several connections between each pair of vertices, it is called a multigraph. Note that a multigraph is a special case of valued graph where all edges between a pair of vertices are aggregated to a unique edge with value counting the original number of edges between the pair. Other types of variables can be associated to the edges such as categorical variables for instance. They are commonly used to give the type of relationship between nodes. In the social framework, real networks are often made of edges representing the so called social interactions. These interactions usually take the form of documents such as articles, emails, text messages, posts, etc. In this scenario, an edge is associated to a collection of texts made of words, which are recorded. We call the graphs made out of these data sets textual graphs.

Covariates Given a network, extra information on the vertices and / or edges can be available. For instance, Figure 2.7 is made out of nodes for which groups indicated by colors are given. The groups can be characterized by categorial variables on the vertices. Other types of covariate information can be provided with continuous variables for instance. Thus, the Florentine business network given in Figure 2.6 represents the business binary relations among 16 Renaissance families. Three quantitative node covariates are given for each family, namely the family's net wealth in 1472, the family's number of seats on the civic councils held between 1282 and 1344, and the family's total number of business and

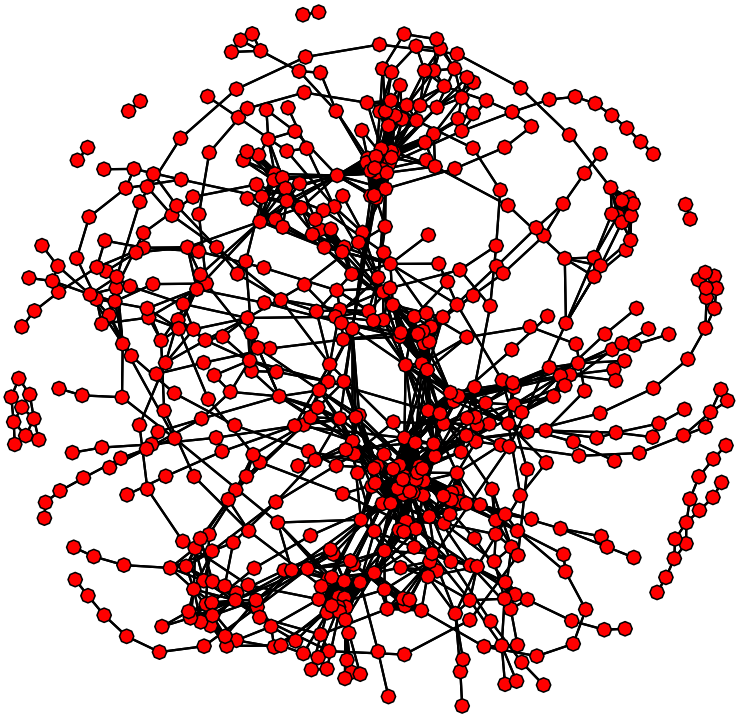


Figure 2.3: The metabolic network of bacteria Escherichia coli (Lacroix et al., 2006). Nodes of the undirected network correspond to biochemical reactions, and two reactions are connected if a compound produced by the first one is a part of the second one (or vice-versa).

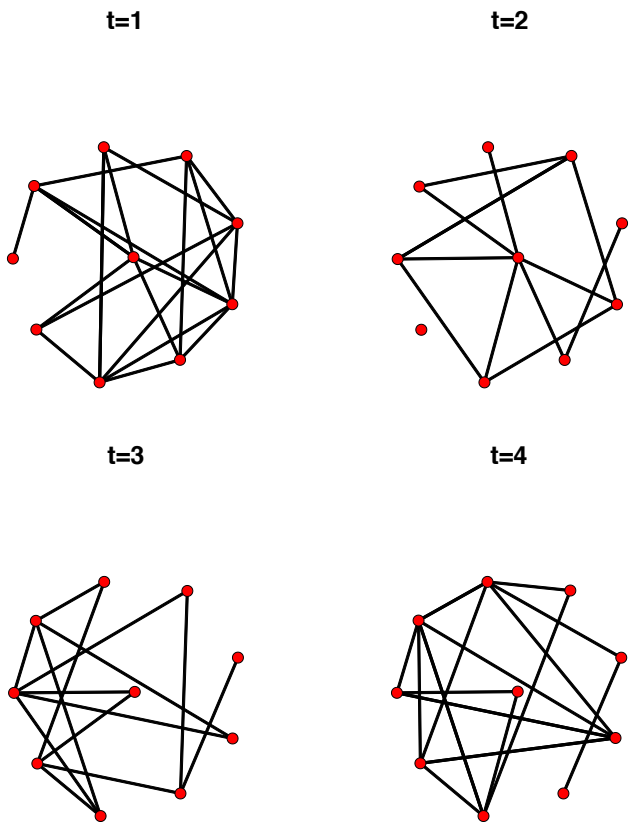


Figure 2.4: Simulated data set of a dynamic network showing the evolution of connections between 10 nodes for 4 different time-points.

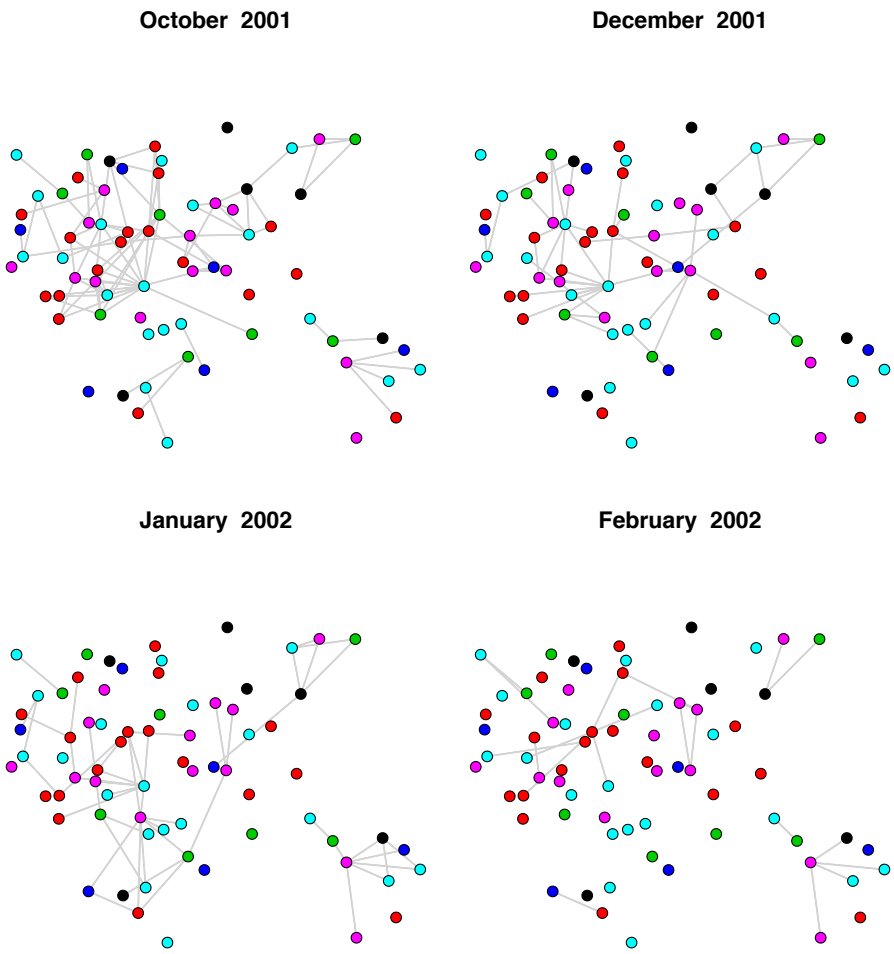


Figure 2.5: Dynamic network between 70 Enron employees for 4 months before and after the bankruptcy of the company. The Enron data set, describes the exchange of emails among individuals who have worked for the Enron company.

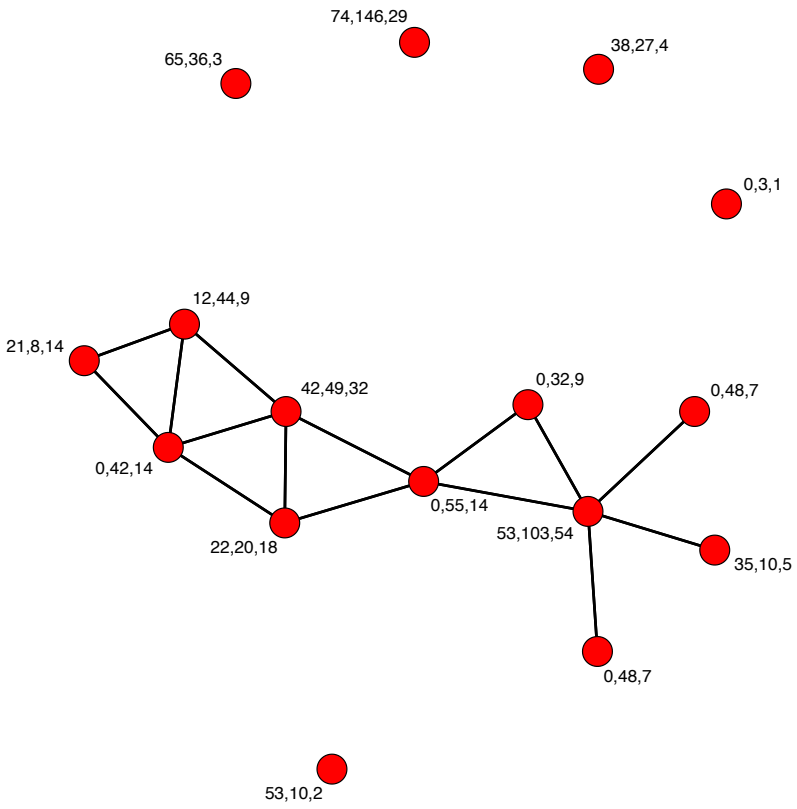


Figure 2.6: Florentine business network describing the business ties between 16 Renaissance families. The vector of covariates for each node is provided.

marriage ties in the entire data set. In some cases, covariate information on the edges is also provided.

Hypergraphs To conclude, let us emphasize that graphs can be extended by allowing edges to connect not exclusively pairs of nodes, but any number of vertices. The corresponding mathematical object is called hypergraph in the literature. Hypergraphs are for instance of interest when describing all the authors of scientific papers. Rather than considering a series of edges to model the pairwise relationships between all the authors of a paper, a unique hyperedge can be taken into account, connecting all the authors.

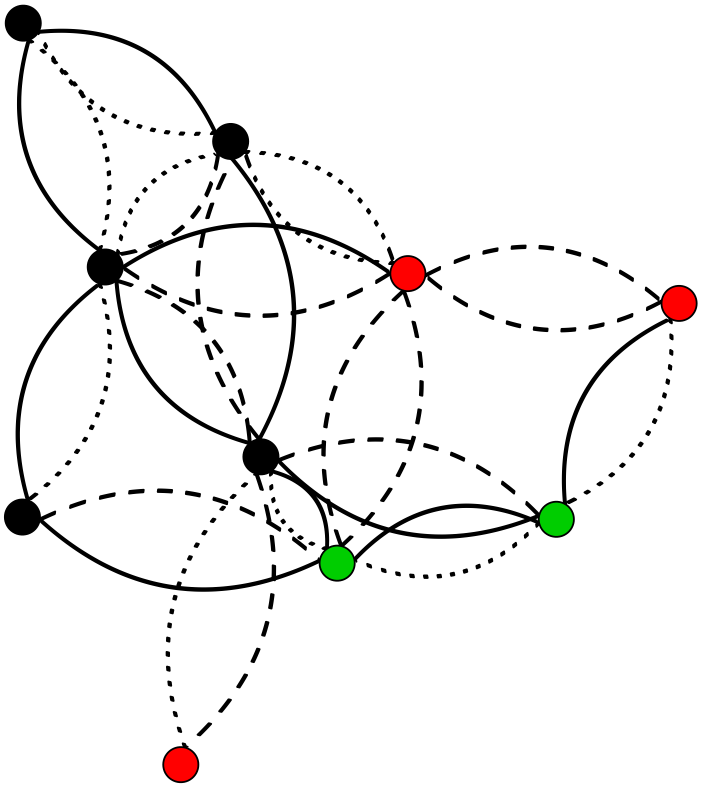


Figure 2.7: An undirected network with 10 nodes and 13 edges having different types, as indicated by their colors and line styles.

2.1.2 Encoding

Many approaches exist to encode graphs as data structures, the two most common ones being as a list of edges and as an adjacency matrix. Other techniques include the use of incidence matrices or successor lists. Note that the latter are essentially only used in operational research, to deal with flows, and are outside the scope of this thesis.

Let us consider a graph $\mathcal{G}$, characterized by a set of vertices denoted by $V(\mathcal{G})$ with N nodes, as well as a set of edges denoted by $E(G)$. The edge list coding simply consists in recording all the edges in $E(G)$ as a list, each element being an edge of $E(G)$. The key advantage of such approach is that only the presence of edges is recorded. The non edges are not taken into account. Conversely, an adjacency matrix stores information for every pair of nodes. This $N \times N$ matrix, denoted by X here, satisfies

$$X_{ij} = \begin{cases} 1 \text{ if node } i \text{ connects to node } j \\ 0 \text{ otherwise.} \end{cases}$$

As an illustration, the network with $N = 9$ nodes, provided in Figure 2.8, can be encoded with the following adjacency matrix and edge list $\{X_{13}, X_{14}, X_{21}, X_{34}, X_{38}, X_{48}, X_{59}, X_{65}, X_{78}, X_{79}, X_{84}\}$.

$$X = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Note that $X_{ii} = 0, \forall i$ and therefore the network does not have any self-loops, that is the connection of node to itself. The adjacency matrix is made of 9 ones corresponding to 11 edges.

2.1.3 Indicators

Several indicators are commonly used to characterize networks. They are also of interest when comparing the global structures of networks.

- number of edges : in this thesis, the number of edges is denoted by m. In the case of an undirected network, it is given by $m = 0.5 \sum_{i,j}^{N} X_{ij}$ and simply $m = \sum_{i,j}^{N} X_{ij}$ for a directed network.

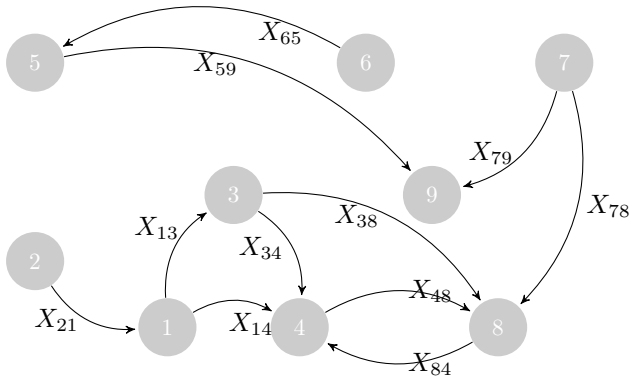


Figure 2.8: An example for a directed network with 9 nodes and 11 edges, in which we label each edge with the component of the adjacency matrix X.

- degree of a vertex : two vertices are called adjacent if they share a common edge, therefore the neighborhood $N(v)$ of a vertex v in a graph $\mathcal{G}$ is the set of vertices adjacent to v. Furthermore, we can characterize each node v by its degree $deg(v)$ which is the number of edges to which v is connected. In other words, the degree of a vertex is the total number of vertices adjacent to the vertex, $deg(v) = |N(v)|$. It can be easily derived from the adjacency matrix X,

$$deg(v) = \sum_{i=1}^{N} X_{iv} + \sum_{i=1}^{N} X_{vi},$$

in the directed case,

$$deg(v) = \sum_{i=1}^{N} X_{iv},$$

in the case of an undirected network.

- density of a network : the density of a network can easily be obtained from the number m of edges present. It is given by

$$\delta(\mathcal{G}) = \frac{m}{L},$$

where L is the number of potential connections, depending on the number N of vertices and the type of network considered. Thus, in the case of networks with self-loop, we have L is given by:

$$L = \begin{cases} N^2 \text{ if directed} \\ N(N+1)/2 \text{otherwise,} \end{cases}$$

and if the network has no self loop,

$$L = \begin{cases} N(N-1) \text{if directed} \\ N(N-1)/2 \text{otherwise.} \end{cases}$$

- Clustering coefficient: the clustering coefficient (Cc) of a vertex represents the probability that the neighbors of a vertex are also connected to each other. That is to say, the clustering coefficient is the probability for two vertices i and j to connect to a third vertex k, such as:

$$\begin{aligned} \text{Cc} &= p(X_{ij}X_{ik}X_{ki} = 1 | X_{ik}X_{jk} = 1) \\ &= \frac{\text{number of pairs of neighbors connected by edges}}{\text{number of pairs of neighbors}}. \end{aligned}$$

2.1.4 Proprieties of real networks

Interestingly, most real networks have been shown to share some properties (Albert et al., 1999; Broder et al., 2000; Dorogovtsev et al., 2000) that we briefly recall in the following.

- **Sparsity:** The number of edges is linear in the number of vertices.
- **Existence of a giant component:** Connected subgraph that contains a majority of the vertices.
- **Heterogeneity:** A few vertices have a lot of connections while most of the vertices have very few links. The degrees of the vertices are sometimes characterized using a scale free distribution (for instance see Barabasi and Albert, 1999).
- **Preferential attachment:** New vertices can associate to any vertices, but "prefer" to associate to vertices which already have many connections.
- **Small world:** The shortest path from one vertex to another is generally rather small.

2.2 Basic of clustering

Clustering and classification are both fundamental tasks in Data Science. Classification is used mostly as a supervised classification method and clustering for unsupervised classification when the class information is missing. The clustering is the process which seeks to divide a data set into homogeneous groups. This process is based on information which is found in the data that describes the objects and their relationship. The goal is to discover as many similarities as possible between the members within a group and as many dissimilarity as possible between groups. More specifically, cluster analysis tries to identify homogeneous groups in a given data set. For example, in biology, cluster analysis can be used for clustering proteins on the based on their characteristics.

In this section, we will focus on the unsupervised classification and we consider three simple and important techniques to introduce the concept of cluster analysis, namely, *hierarchical clustering* (based on the Agglomerative and divisive algorithms), *partitional clustering* (based on the K-Means algorithm) and *mixture model* clustering.

2.2.1 Partitional Clustering

The best examples of this family of clustering are K-means and K-medoids (also known as partition around medoids (PAM)). The K-means clustering was proposed by MacQueen et al. (1967) and intends to partition N objects into Q clusters in which each object belongs to the cluster with the nearest mean. Here, we have supposed that the number Q of clusters is fixed. In Section 2.6.1, we will see how Q can be estimated from the data. Let us consider a continuous data set $\{x_1, x_2, \ldots, x_N\}$ consisting of N observations of a random d-dimensional vector. Our goal is to partition the data into Q disjoint clusters $\{P_1, \cdots, P_Q\}$ such that $P_r \cap P_l = \emptyset$, where the prototype of any cluster $q \in \{1, \ldots, Q\}$ is often a centroid, i.e the average (mean) of all points in the cluster noted η_q. When the data has categorical variable, the prototype is often a medoid i.e, the most representative point of a cluster. For each point $\{x_n\}_{1,\ldots,N}$, we introduce a corresponding indicator variable Z_{nq} such as it equals to 1 if x_n belongs to cluster q and 0 otherwise. Then, the objective of K-means clustering is to find values of the $\{Z\}$ and $\{\eta\}$ which minimize the sum of squares of the distances between each point and its closest prototype (η):

$$J = \sum_{i=1}^{N} \sum_{q=1}^{Q} Z_{iq} \|x_i - \eta_q\|^2.$$

To start this algorithm (Algorithm 1), we select Q random points as cluster centers. Then, in the first step, we minimize J with respect to Z while the set $\{\eta_1, \ldots, \eta_Q\}$ is fixed. The corresponding computational cost is $O(QNd)$. In the second step, J is minimized with respect to the set of $\{\eta\}_{q=1,\ldots,Q}$ keeping Z fixed. The time required here for calculating the centroids is $O(Nd)$. We repeat the steps 1 and 2 until the same points are assigned to each cluster in two consecutive rounds. We note that, in these two steps, we calculate the centroid or mean of all objects in each cluster and assign objects to their closest cluster center according to the Euclidean distance function. An example of the clustering result of K-means is shown in Figure 2.9.

2.2.2 Hierarchical Clustering

Hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom, i.e. a tree of clusters, also known as a dendrogram (see Figure 2.10). Hierarchical clustering methods can be categorized into agglomerative (bottom-up) and divisive (top-down) (Jain and Dubes, 1988; Kaufman and Rousseeuw, 1990). In order to decide which clusters should be combined (for agglomerative), or whether a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, this is achieved by the use of an appropriate metric (a distance measure between pairs of observations). Distances between objects can be visualized in many simple yet clear ways. For example, the initial

Algorithm 1: The basic procedure of K-means.

INITIALIZATION
Step 0 : Random initialization of Q clusters (η_0)

OPTIMIZATION

$\eta_q^{old} \leftarrow \eta_q^{new}, \forall q$

repeat
step 1 : Assign each data object to its nearest cluster η_q, such as
for i in 1: N **do**
q= $\text{argmin}_l \|x_i - \eta_l^{old}\|^2$
$Z_{iq} \leftarrow 1$
$Z_{il} \leftarrow 0, \forall\, l \neq q$
step 2 : Update the centroid of each changed cluster
until there is no change in any cluster

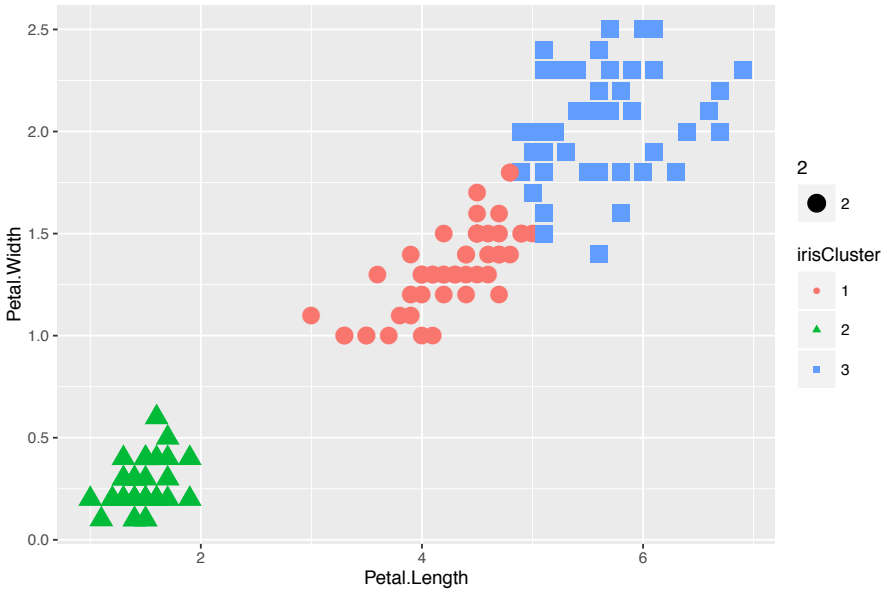


Figure 2.9: An example of the K-means clustering of the "iris" data set. This data contains the features of three differents species of flower. The results demonstrate that that Petal.Length and Petal.Width were similar among the same species but varied considerably between different species.

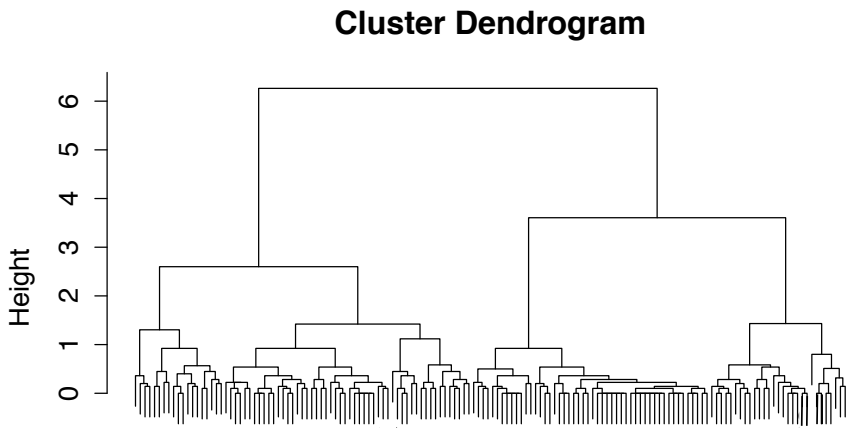


Figure 2.10: An example of the dendrogram of the "iris" data set. This data contains the features of three differents species of flower. The results demonstrate that that Petal.Length and Petal.Width were similar among the same species but varied considerably between different species.

distance measure between the initial elements may be the Euclidean distance:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2},$$

or any other distance such as the squared Euclidean distance, the rectangular distance, the maximum distance, the Chi-2 distance, etc.

Agglomerative clustering has an $O(n^2 \log n)$ complexity and usually uses as input a dissimilarity matrix on the initial elements (Algorithm 2). This algorithm starts with one-point (singleton) clusters and recursively merges two or more most appropriate clusters. In this method, we first assign each observation to its own cluster. Then, we compute the similarity (e.g., distance) between each pair of clusters, and join the two most similar ones. Then, we repeat steps two and three until there is only a single cluster left. This algorithm is shown below (see Figure 2.10).

A divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster. The process continues until a stopping criterion (frequently, the requested number k of clusters) is achieved. In this method we assign all of the observations to a single cluster and then partition the cluster to two least similar clusters. Finally, we proceed recursively on each cluster until there is one cluster for each observation.

Algorithm 2: Specifications of all agglomerative hierarchical clustering methods.

Step 1 : start with N clustering: basically each object is a cluster and calculate the proximity matrix for N clusters

repeat

step 2 : Find minimum distance in the proximity matrix and merge the two clusters with the minimal distance

step 3 : Update the proximity matrix

until all objects are in one cluster

2.2.3 Mixture model clustering

Mixture model clustering is another family of clustering methods, which has attracted more and more attention recently. It is considered as the probabilistic approach where the data is supposed to be a sample independently drawn from a mixture model of several probability distributions McLachlan and Basford (1988); McLachlan and Peel (2004). Mixture model clustering can be classified into two large groups, namely finite mixture models (parametric models) and infinite models (non parametric models). Here we will interested to the finite mixture models for clustering a data.

The finite mixture model of probability distributions assumed that the data containing Q homogeneous sub-populations (groups) called components (see Figure 2.11). Therefore, the total populations is a mixture of these Q groups. Let us consider $X = \{x_1, \ldots, x_N\}$ a sample of N random variables independent, identically distributed. Each is variable assumed to be distributed according to a mixture of Q components, of density f, such as:

$$f(x) = \sum_{k=1}^{K} \alpha_k f_k(x), \tag{2.1}$$

where, the coefficient α_k, called mixing proportions or weight components, such as $0 < \alpha_k < 1$ and $\sum_{k=1}^{K} \alpha_k = 1$. Regardless of the distribution of f, the mixture model in (2.1) can be seen as the result of a marginalization over a latent variable (Z), where Z is a binary K-vector, assumed to be draw from a multinomial distribution of parameter α, such that:

$$\begin{aligned} p(Z_i|\alpha) &= \mathcal{M}(Z_i; 1, \alpha = (\alpha_1, \ldots, \alpha_K)) \\ &= \prod_{q=1}^{Q} \alpha_q^{Z_{iq}}, \end{aligned}$$

where, $Z_{iq} = 1$ means that observation i belongs to class q.

Figure 2.11: An example of the finite Gaussian mixture clustering fitted via EM algorithm of the "iris" data set. This data contains the features of three differents species of flower. The results demonstrate that that Petal.Length and Petal.Width were similar among the same species but varied considerably between different species.

Gaussian mixture models Let us now consider that we have a mixture model with the Gaussian density defined on R^d with Q mean vectors $\nu_q = E(X|Z_q = 1)$ and covariance matrices Σ_q, such that, each Gaussian density $N(x_i; \theta_q)$ is called a component of the mixture and has it own mean ν_q and covariance Σ_q. In this case, the density function is given:

$$f(x_i; \nu, \Sigma) = N(x_i; \nu, \Sigma) = \sum_{q=1}^{Q} \frac{1}{(2\Pi)^{d/2}|\Sigma_q|^{1/2}} \exp\Big(-\frac{1}{2}(x_i - \nu_q)^T \Sigma_q^{-1}(x_i - \nu_q)\Big),$$

we denote $\theta = \{\theta = (\nu_q, \Sigma_q)\}_{1,\ldots,Q}$ which defines the set of model parameters where the vector $\nu_q \in R^d$ denotes the mean of the component q, in which $\nu_q = E(X|Z_q = 1)$ and the matrix $\Sigma_q \in R^{d\times d}$ defines the covariance of the component q. Therefore, the conditional distribution of x_i given a value for Z_i is a Gaussian distribution, such that,

$$p(x_i|Z_{iq} = 1, \theta_q) = N(x_i; \nu_q, \Sigma_q),$$

and the distribution of the all data can be written in the following forms:

$$p(x|Z) = \prod_{q=1}^{Q} N(x_i; \nu_q, \Sigma_q)^{Z_q}.$$

Finally, the joint distribution is given by the marginalization of $p(x_i|\alpha, \theta)$ over all possible vectors Z_i, such as:

$$\begin{aligned} p(x_i|\alpha, \theta) &= \sum_{Z_i} p(x_i, Z_i|\alpha, \theta) \\ &= \sum_{Z_i} \prod_{q=1}^{Q} p(x_i, |Z_i = 1, \nu_q, \Sigma_q) p(Z_i|\alpha_q) \\ &= \sum_{Z_i} \prod_{q=1}^{Q} \Big(\alpha_q N(x_i; \nu_q, \Sigma_q))^{Z_{iq}}\Big). \end{aligned}$$

It is worth noticing that the K-means algorithm can be viewed as a Gaussian mixture model with spherical covariance matrices and equal proportions. To obtain the estimation of mixture model parameters, we can apply the standard approach in machine learning which is the EM algorithm (mentioned in Section 3).

2.2.4 Graph clustering

In the context of graph clustering, the data sets are often presented in the form of edge lists or adjacency matrices. The goal is then to analyze the connections and other information provided in order to build clusters of vertices sharing common features. Many types of clusters of nodes can be taken into account. Looking for specific types of clusters may require to impose strong constraints on the models and the corresponding inference techniques. In the following, we provide two examples of structures which are often considered in real applications.

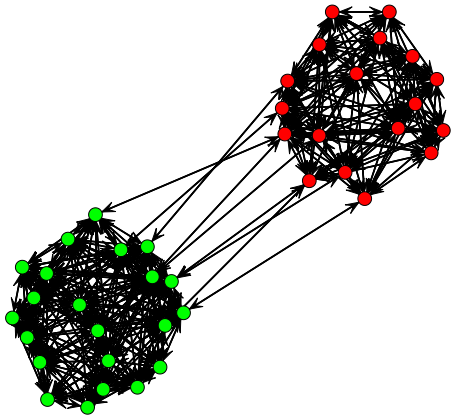


Figure 2.12: An example of an undirected network with 40 nodes showing the structure of two partially isolated communities represented in red and green.

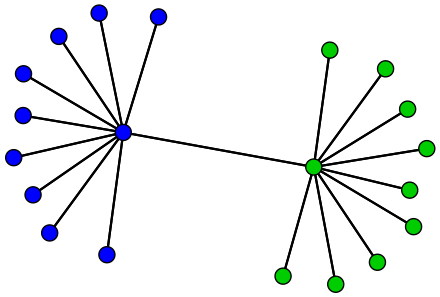


Figure 2.13: An example of the star cluster of an undirected graph with 20 nodes.

Community clusters A community is made out of nodes which exhibit a transitivity property such that nodes of the same community are more likely to be connected. In other words, communities of nodes have a higher density of edges inside a group rather than between groups (see Figure 2.12).
We note that a graph can be also characterized using the notion of clique which is a particular case of community. A clique is a subset of vertices of an undirected graph, where each vertex is adjacent to each other. In the other words, all nodes in this subset are interconnected. For instance, in biology, we use cliques as a method of abstracting pairwise relationships, such as, gene similarity where the goal is to establish an edge between two genes having similar profile. The maximal clique is the largest subset of vertices in which each point is directly connected to every other vertex in the subset.

Disassortative mixing or stars Unlike the community pattern, the star pattern consists of one central node, so called hub, and a set of nodes which are connected to it. For instance, workstations are directly connected to a common central computer. The degree distribution of the nodes in this kind of cluster is heavily skewed and the probabilities of connections within a group are lower than the probabilities of connections between groups (see Figure 2.13).

2.3 Clustering models for static graphs

In the previous sections, we have introduced some basic definitions of graph theory and cluster analysis. In this section, we now concentrate on describing the existing methods for the clustering of nodes in static networks.

The concept of the clustering of nodes has different meanings in the literature. Among these notions, we note especially White et al. (1976) who have extensively studied this problem, both empirically and theoretically through a transitivity of relations. Transitivity means here, that two actors that have ties with a third actor are more likely to be tied than actors that do not. For example, if we observe $i \to j$ and $j \to k$, then i and k are more likely to be connected. Also, clustering can be defined based on the homophily by attributes, which was studied by Freeman (1996); McPherson et al. (2001) which explained that ties are often more likely to occur between actors that have similar attributes than between those who do not.

Overall, two families of approaches can be highlighted, depending on the type of structure they aim at uncovering and the type of edges analyzed. Thus, most techniques look for so called communities where vertices within a community are more likely to be connected than vertices of different communities. They are widely used in social sciences for instance. Alternatives methods look for heterogeneous structures which include hubs of star patterns. They can also be employed in order to look for communities, but not only. Although some attempts have been made to extend the concept of community to networks with categorical edges (see Labiod and Bennani, 2011, for instance) or to multi graphs, this concept is usually associated with networks with binary edges. Thus, in

this manuscript, the term cluster will be used when looking for heterogeneous structures in binary networks and / or networks made out of non binary edges.

2.3.1 Community structure

Most clustering algorithms looking for communities involve optimization techniques from physics and computer science. Only a small portion of them take a statistical point of view and rely on random graph models to characterize the generative (random) construction of the graph.

Modularity score The modularity score was proposed by Newman and Girvan (2004). It is given by

$$\text{mod} = \sum_{q=1}^{Q} (e_{qq} - a_q^2),$$

where e_{ql} is the fraction of edges in the network that link vertices in community q to vertices in community l. Moreover, $a_q = \sum_{l=1}^{Q} e_{ql}$ denotes the fractions of edges that connect to vertices of community q. Maximizing this criterion induces a search for clusters where the number of edges within each cluster is unexpectedly large with respect to a null model. A long series of heuristics have been proposed for this purpose. For instance, the algorithm of Newman and Girvan (2004) allows the iterative removal of edges using one of a number of possible betweenness measures. The criterion is then computed for all the divisions, and a division is chosen such that the modularity score is maximized. However, (Bickel and Chen, 2009) showed that these algorithms optimizing the modularity score are (asymptotically) biased and tended to lead to the discovery of an incorrect community structure, even for large graphs.

Latent position cluster model The most popular random graph model considering transitivity and dealing with communities is the latent position cluster model (LPCM) proposed by Handcock et al. (2007), as an extension of the latent space model of Hoff et al. (2002).

Each actor is given a random latent position Z_i in $\mathbb{R}^p$ by sampling from a finite Gaussian mixture model, each component representing a community of nodes

$$Z_i \sim \sum_{q=1}^{Q} \alpha_q \mathcal{N}(\nu_q, \sigma_q^2 \mathbf{I}).$$

The vector $\alpha = (\alpha_1, \ldots, \alpha_Q)$ denotes the cluster proportions. Moreover, each multivariate normal distribution has a different mean vector ν_q as well as spherical covariance matrix $\sigma_q^2 \mathbf{I}$. Then, the presence or absence of an edge between each pair (i, j) of vertices is explained by the corresponding latent vectors Z_i and Z_j. Please note that the model proposed in the original paper allows to deal with covariates on the edges. Thus, denoting y_{ij} the set of covariates for the pair

(i, j), X_{ij} is assumed to be drawn from a Bernoulli distribution:

$$X_{ij}|Z_i, Z_j, y_{ij} \sim \mathcal{B}(p_{ij}),$$

where

$$\text{logit}(p_{ij}) = \beta_0 + \beta^T y_{i,j} - |Z_i - Z_j|.$$

Therefore, the distance between the vectors Z_i and Z_j is key to the construction of an edge. The closer the vectors are, the larger is the probability of a connection.

In a generative perspective, all the latent positions are first sampled independently. Then, given the positions, the edges are drawn independently. Standard inference techniques for the LPCM model include Monte Carlo Markov Chain (MCMC) (Krivitsky and Handcock, 2008) and variational expectation maximization (VEM) (Salter-Townshend and Murphy, 2009).

2.3.2 Cluster structures

In this section, we now consider more flexible random graph models, capable of retrieving various patterns of connections and / or deal with non necessarily binary edges.

Stochastic block model

The stochastic block model (SBM) (Wang and Wong, 1987; Nowicki and Snijders, 2001) is a flexible random graph model which concentrates on the classification of nodes in a network depending on their connection probabilities. It is based on a probabilistic extensions of the method applied by White et al. (1976) on Sampson's famous monastery (Fienberg and Wasserman, 1981b). It assumes that each vertex belongs to a latent group, and that the probability of a connection between a pair of vertices depends exclusively on their groups. As such, it generalizes the Erdös-Rényi model (Erdös and Rényi, 1959) which supposes that two vertices taken at random connect with an homogeneous probability. Because no specific assumption is made on the connection probabilities, various types of structures of vertices can be taken into account by SBM. This model has indeed the ability to characterize clusters such as communities and stars or disassortative clusters. While SBM was originally developed to analyze mainly binary networks, many extensions have been proposed since to deal, for instance, with valued edges (Mariadassou et al., 2010), categorical edges (Jernite et al., 2014) or to take into account prior information (Zanghi et al., 2010; Matias and Robin, 2014). Note that other extensions of SBM have focused on looking for overlapping clusters (Airoldi et al., 2008; Latouche et al., 2011).

Let us consider an undirected graph $\mathcal{G}$, characterized by its $N \times N$ binary adjacency matrix X, which is assumed not to have any self loop, therefore the diagonal entries X_{ii} are all zeros. The SBM model is a mixture model for graphs which supposes that vertices are spread into Q classes with prior probabilities $\alpha = (\alpha_1, \ldots, \alpha_Q)$ where α_q is the proportion of the qth cluster. Furthermore, each vertex i is associated to a unique cluster, such that $Z_{iq} = 1$ if vertex i

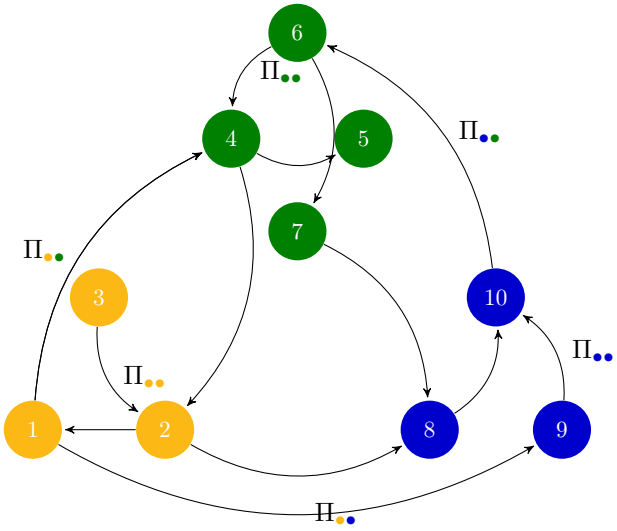


Figure 2.14: An example for the matrix of connection probabilities between clusters (Π) in the SBM model. The network is made of 10 nodes split into $K = 3$ clusters (indicated by the colors).

belongs to cluster q, and 0 otherwise. The vector $Z_i = (Z_{i1}, \ldots, Z_{iQ})$ is then assumed to be drawn from a multinomial distribution:

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

Thus, as for Gaussian mixture models (Section 2.2.3), Z_i is a binary vector of size Q, with a unique component set to one, such that $\sum_{q=1}^{Q} Z_{iq} = 1$.

On the other hand, the presence of a link between nodes i and j is assumed to be sampled from a Bernoulli distribution, depending on the latent vectors Z_i and Z_j:

$$X_{ij} | Z_{iq} Z_{jr} = 1 \sim \mathcal{B}(\Pi_{qr}),$$

where Π is a $Q \times Q$ matrix of connection probabilities between clusters (illustrated in Figure 2.14) with $\Pi_{qr} \in [0, 1]$. Depending on the form of this matrix, various type of structure can be taken into account. For instance, an affiliation network made of communities can be defined by the following matrix Π

$$\Pi = \begin{pmatrix} \lambda & \epsilon & \cdots & \epsilon \\ \epsilon & \lambda & \cdots & \epsilon \\ \vdots & \vdots & \vdots & \vdots \\ \epsilon & \epsilon & \ldots & \lambda \end{pmatrix},$$

where λ is a high probability of connections between nodes within the cluster, while ϵ is a low probability of connections between groups.

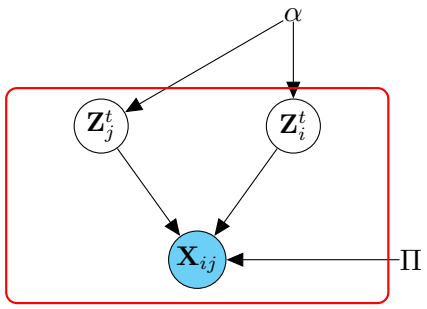


Figure 2.15: Graphical representation of the stochastic block model.

According to this model, the latent variables $Z_1, ..., Z_N$ are iid and given this latent structure, all the edges are supposed to be independent. The SBM model is then defined by the following joint distribution, and the corresponding graphical model is provided in Figure 2.15.

$$p(X, Z|\Pi, \alpha) = p(X|Z, \Pi)p(Z|\alpha),$$

where

$$\begin{aligned} p(X|Z, \Pi) &= \prod_{i \neq j}^{N} p(X_{ij}|Z_i Z_j, \Pi) \\ &= \prod_{i \neq j}^{N} \prod_{q,r}^{Q} \mathcal{B}(X_{ij}|\Pi_{qr})^{Ziq Zjr} \\ &= \prod_{i \neq j}^{N} \prod_{q,r}^{Q} (\Pi_{qr}^{X_{ij}} (1 - \Pi_{qr})^{X_{ij}})^{Ziq Zjr}, \end{aligned}$$

and

$$\begin{aligned} p(Z|\alpha) &= \prod_{i=1}^{N} \mathcal{M}(Z_i; 1, \alpha) \\ &= \prod_{i=1}^{N} \prod_{q=1}^{Q} \alpha_q^{Z_{iq}}. \end{aligned}$$

Consequently, the SBM model is described by its set of latent variables Z and (α, Π) as parameters. To perform inference on real data sets, many methods have been proposed in the literature such as: VEM (Daudin et al., 2008), variational Bayes EM (VBEM) (Latouche et al., 2012), or Gibbs sampling (Nowicki and Snijders, 2001). Some of these methods will be explained in Section 2.6.

Mixed membership stochastic block model

The mixed membership SBM (MMSBM) model was proposed by Airoldi et al. (2006). It considers a hierarchy of probabilistic assumptions about how objects interact with one another. It can be seen as an extension of the SBM model by allowing the vertices to belong to multiple clusters. We consider here a directed graph without self loops.

Each node i is first associated with a vector β_i of size Q such as:

$$\beta_i \sim Dirichlet(\alpha).$$

The parameter $\beta_i = (\beta_{i1}, \ldots, \beta_{iQ})$ is such that $\sum_{q=1}^{Q} \beta_{iq} = 1$ where β_{iq} is the probability of node i to be in cluster q. By construction, several components of the vector β_i can be different from zero and therefore i is allowed to belong to several clusters simultaneously. Then, for the pair (i, j) of vertices, two latent variables $Z_{i\to j}$ and $Z_{i\leftarrow j}$ are drawn:

$$Z_{i\to j} \sim \mathcal{M}(1, \beta_i),$$

and

$$Z_{j\leftarrow i} \sim \mathcal{M}(1, \beta_j).$$

Both vectors are binary and contain a unique one. Thus, in the relationship from vertex i and vertex j, i and j are associated to specific clusters. The cluster of i might change when looking at a different pair of vertices from or to i. Finally, knowing $Z_{i\to j}$ and $Z_{i\leftarrow j}$, the presence of an edge between i and j is supposed to be sampled from a Bernoulli distribution with probability:

$$X_{ij} | Z_{i\to j,q} Z_{j\leftarrow i,l} = 1 \sim \mathcal{B}(\Pi_{ql}).$$

The $Q \times Q$ matrix Π of connection probabilities is similar to the SBM connection probability matrix.

So, the MMSBM model has two set of latent variables (Z, β). The joint probability of the data X and these variables is given by

$$P(X, Z, \beta | \alpha, \Pi) = \prod_{i \neq j}^{N} p(X_{ij} | Z_{i\to j}, Z_{j\leftarrow i}, \Pi) p(Z_{i\to j} | \beta_i) p(Z_{j\leftarrow i} | \beta_j) \prod_{i=1}^{N} p(\beta_i | \alpha).$$

Airoldi et al. (2006, 2008) used a VEM to approximate the posterior distributions of the latent variables and to estimate the model parameters.

Random subgraph model

So far, we have basically presented random graph models aimed at modeling binary edges. In this section, we describe an extension of the SBM model, called the random subgraph model (RSM), as proposed by Jernite et al. (2014). The RSM model aims at modeling categorical edges using prior knowledge of a partition of the network into different subgraphs. Each subgraph is assumed to be

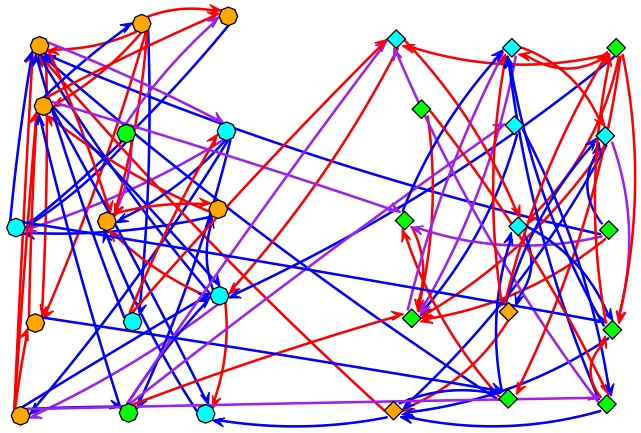


Figure 2.16: Example of an RSM network.

made of latent clusters which have to be inferred from the data in practice. Then, the vertices are connected with probabilities depending only on the subgraphs, whereas the edge type is assumed to be sampled conditionally on the latent groups.

An example of an RSM network is given in Figure 2.16. The node forms indicate the (known) partition of the network into $S = 2$ subgraphs. Moreover, the edge types $X_{ij} \in \{0, ..., C\}$ ($C = 3$) are given by the edge colors. Within each subgraph, the $Q = 3$ clusters are indicated by the node colors. As for the SBM and MMSBM models, the construction of the adjacency matrix is assumed to depend on latent clusters. Thus, each node i is first associated with an unobserved group among Q with a probability depending on s_i, where s_i indicates the subgraph of vertex i:

$$Z_i \sim \mathcal{M}(1, \alpha_{s_i}).$$

The vector α_{s_i} denotes the cluster proportions for the corresponding subgraph. Secondly, the presence of an edge between two nodes i and j is characterized by an observable variable A_{ij}, such that $A_{ij} = 1$ is there exists a typed relation between i and j, 0 otherwise. The edge type is then encoded by the observable variable X_{ij} which takes its values in a finite set $\{0, 1, ..., C\}$.

The variable A_{ij} is supposed to be drawn from a Bernoulli distribution depending on the subgraphs s_i and s_j only:

$$A_{ij} \sim \mathcal{B}(\gamma_{s_i,s_j}).$$

Then, if an edge is present between i and j, X_{ij} is sampled from a multinomial distribution with probabilities depending on the latent clusters.

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

where $\Pi_{ql} \in [0,1]^{C+1}$ and $\sum_{c=0}^{C} \Pi_{ql}^{c} = 1$.

Therefore, the model is defined through the following joint distribution:

$$\begin{aligned} p(X, A, Z|\alpha, \gamma, \Pi) &= p(X, A|Z, \gamma, \Pi)p(Z|\alpha) \\ &= p(X|A, Z, \Pi)p(A|\gamma)p(Z|\alpha). \end{aligned}$$

In the original paper, the inference is performed using a VBEM algorithm.

2.4 Clustering models for dynamic graphs

In Section 2.1, we defined some approaches capable of analyzing static networks. In this section, we now wish to extend models from the previous section to the dynamic framework. First, we introduce the dynamic mixed membership stochastic block model (dMMSBM), which is an extension of the static MMSBM model. Then, we present the dynamic SBM (dSBM) model, which extends the static SBM model.

2.4.1 Context and notations

We are now provided with a dynamic graph $\mathcal{G}$ where edges can appear or vanish over time. Conversely, the vertex set is assumed to be fixed. $\mathcal{G}$ is then defined through a series of T networks $\mathcal{G} = \{\mathcal{G}^{(t)}\}_{t=1}^{T}$ where $\mathcal{G}^{(t)}$ is a (fixed) graph at time t. More precisely, $\mathcal{G}^{(t)}$ is the (aggregated) graph of all the connections that occurred during time frame t. In other words, in the binary case, two vertices i and j in $\mathcal{G}^{(t)}$ are connected and the presence of the edge (i, j) is recorded if there was at least one connection between i and j, during t. Furthermore, in the weighted case, the number of interactions for the edge (i, j) is recorded. As such, the dynamic network $\mathcal{G}$ can be seen as a time series of networks. Each graph $\mathcal{G}^{(t)}$ is then represented by its $N \times N$ adjacency matrix $X^{(t)}$. Thus, in the binary case $X_{ij}^{(t)} = 1$ if the edge (i, j) is present in the graph $\mathcal{G}^{(t)}$, 0 otherwise. Moreover $X_{ij}^{(t)}$ is set to the number of interactions that occurred during time frame t, in the weighted case. Note that no self loops are considered here and therefore $X_{ii}^{(t)} = 0$, $\forall\, i, t$. In this context, clustering the data means clustering the vertices at each time t.

2.4.2 Dynamic mixed membership stochastic block model

The dynamic mixed membership stochastic block model (dMMSBM) is a random graph model for dynamic binary graphs proposed by Xing et al. (2010). The idea at the core of this approach is to extend the MMSBM (Airoldi et al., 2008) model by including a state-space model to characterize the evolution of the latent space variables. Below, the model is presented in the case of directed graphs, where connections are oriented.

First, a vector $\beta_i^{(t)}$ is considered for each node i at time t in the graph. This vector characterizes the probabilities for the node to belong to the various

clusters. By construction, a node can belong to multiple clusters, at each time t. Contrary to MMSBM, $\beta_i^{(t)}$ is not directly sampled from a Dirichlet distribution, but rather obtained from a logistic normal one. Thus, each term $\beta_{iq}^{(t)}$ in $\beta_i^{(t)}$ is such that:

$$\beta_{iq}^{(t)} = \exp(\gamma_{iq}^{(t)} - C(\gamma_i^{(t)})), \forall q, i, t,$$

where $C(\gamma_i^{(t)}) = \log(\sum_{q=1}^{Q} \exp(\gamma_{iq}^{(t)}))$ is a normalization constant. Due to the bijectivity constraint of this ogistic-like transformation, $\gamma_i^{(t)}$ lives in a $(Q-1)$ dimensional space since $\beta_i^{(t)}$ has $(Q-1)$ degrees of freedom. In addition, the first $(Q-1)$ components of the vector $\gamma_i^{(t)}$ are assumed to be Gaussian, with mean $B\nu^{(t)}$ and covariance matrix Σ:

$$\gamma_i^{(t)} \sim \mathcal{N}(B\nu^{(t)}, \Sigma). \tag{2.2}$$

A state-space model is then introduced to encode the evolution of the $\nu^{(t)}$ parameters:

$$\begin{cases} \nu^{(t)} = A\nu^{(t-1)} + \omega \\ \nu^{(1)} = \mu_0 + u, \end{cases}$$

where the noise terms ω and u are supposed independent Gaussians:

$$\begin{cases} \omega \sim \mathcal{N}(0, \Phi) \\ u \sim \mathcal{N}(0, V_0). \end{cases}$$

Note that A, Φ, B, and V_0 are matrices of size $(Q-1) \times (Q-1)$, while μ_0 is a $(Q-1)$-dimensional vector.

This stochastic process allows to model the evolution of the mixed membership vectors over time. The dMMSBM model also aims at characterizing the evolution of the connection probabilities between clusters. Thus, a second state space model is employed and the probability for nodes of clusters q and l to connect at time t is given by the logistic function:

$$\Pi_{ql}^{(t)} = \frac{1}{1 + \exp(-\eta_{ql}^{(t)})}, \forall q, l, t,$$

where

$$\begin{cases} \eta_{ql}^{(t)} = b\eta_{ql}^{(t-1)} + \epsilon_1 \\ \eta_{ql}^{(1)} = \varsigma + \epsilon_2. \end{cases}$$

The random variables ϵ_1 as well as ϵ_2 are Gaussian random variables both drawn from $\mathcal{N}(0, \psi)$.

Finally, at each time t, given all the vectors $\beta_i^{(t)}$ and the probabilities $\Pi_{ql}^{(t)}$, the remaining sampling scheme is similar to the one in MMSBM. Thus, in the connection from vertex i to vertex j, at time t, two binary latent vectors are sampled from multinomial distributions

$$Z_{i \to j}^{(t)} \sim \mathcal{M}(1, \beta_j^{(t)}),$$

and

$$Z_{j\leftarrow i}^{(t)} \sim \mathcal{M}(1, \beta_j^{(t)}).$$

These vectors have similar interpretation as in MMSBM. Then, the presence of an edge is supposed to be drawn according to a Bernoulli distribution:

$$X_{ij}^{(t)} | Z_{i\to j,q}^{(t)} Z_{j\leftarrow i,l}^{(t)} = 1 \sim B(\Pi_{ql}^{(t)}).$$

The dMMSBM model has four sets of latent variables ($\nu = (\nu^{(t)})_t, \gamma = (\gamma_i^{(t)})_{it}, Z = (Z_i^{(t)})_{it}$, $\Pi = (\Pi_{ql}^{(t)})_{qlt}$) and is parameterized by $\theta = (\varsigma, \psi, B, A, b, \Phi, V_0, \Sigma, \eta)$. The inference is made using a VEM algorithm. The joint distribution is given by:

$$P(X, Z, \beta, \Pi, \gamma | \theta) = \prod_{t=1}^{T} \prod_{i,j}^{N} p(X_{ij}^{(t)} | Z_{i\to j}^{(t)}, Z_{j\leftarrow i}^{(t)}, \Pi^{(t)}) p(\Pi^{(t)} | \gamma^{(t)}) p(\gamma^{(t)} | A, \nu, \Sigma)$$

$$p(Z_{i\to j}^{(t)} | \beta_i^{(t)}) p(Z_{j\leftarrow i}^{(t)} | \beta_j^{(t)}) \prod_{i=1}^{N} p(\beta_i^{(t)} | \eta^{(t)}) p(\eta^{(t)} | \varsigma, \psi).$$

2.4.3 Dynamic stochastic block model

One the most recent models suggested for dynamic networks is called the dynamic stochastic block model (dSBM), proposed by Matias and Miele (2016). A dSBM is a combination of a SBM model for its static part, with independent Markov chains for evolution of the node groups through time. This model studies the evolution and characterization of nodes through time, with binary or weighted edges, with a stable connectivity behavior within groups.

To describe dSBM, we will use the notations and main principles of SBM stated in Section 2.3.2. Contrary to the previous section, the T networks provided for the different time frames are assumed to be undirected. Thus, the corresponding adjacency matrices $X^{(t)}$ are symmetric.

The goal of dSBM is to cluster, at each time t, the N nodes into Q latent groups, i.e., find an estimate of the set of latent variables $Z = \{Z_i^{(t)}\}_{1\leq t\leq T, 1\leq i\leq N}$ with values in $\{1, \ldots, Q\}^{NT}$. Here, $Z_i^{(t)}$ is a Q-vector in $\{0, 1\}$ and $Z_{iq}^{(t)} = 1$ if at time t, node i belongs to the class q, and 0 otherwise.

Given the latent groups Z, the T graphs are independent, and at each time t, conditional on $Z^{(t)}$, the edges are assumed independent. On other words, at time t the weighted edge $X_{ij}^{(t)}$ depends only on the latent variables $Z_i^{(t)}$ and $Z_j^{(t)}$. So, once we begin estimating the latent variable Z over time, we analyze independently the T graphs like for SBM models. To this end, Matias and Miele (2016) have proposed independent Markov chains for analyzing the evolution of the groups of nodes through time, as follows: for each time t, an individual i belongs to cluster q with probability α_q, which can change at $(t + 1)$. Then, the process $Z_i = \{Z_i^{(t)}\}_{1\leq t\leq T}$ is supposed an irreducible, aperiodic stationary Markov chain with transition matrix Π, a matrix of connections between groups, and an initial stationary distribution $\alpha = (\alpha_1, \ldots, \alpha_Q)$.

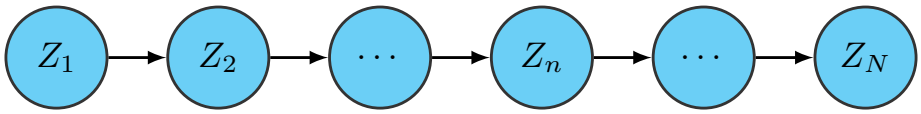


Figure 2.17: The graphical model of a first-order Markov chain.

Furthermore, at time t, and conditional on the latent structure $Z^{(t)}$, the random variables $X_{ij}^{(t)}$ are independent, and the distribution of each X_{ij} depends only on Z_i and Z_j. In the other hand, following Ambroise and Matias (2012), the weighted link between nodes i and j is assumed to be sampled from the following form of distribution:

$$X_{ij}^{(t)}|Z_i^{(t)}Z_j^{(t)} = 1 \sim (1-\beta_{ql}^{(t)})\gamma_0(.) + \beta_{ql}^{(t)}F(.,\gamma_{ql}^{(t)}),$$

where $(F(.,\gamma), \gamma \in \Gamma)$ is a parametric family of distributions with no point mass at 0. This distribution can take many forms, such as, for example, a Gaussian family with unknown mean and variance, a Poisson family on $N \setminus 0$, etc. Since dSBM allows us to study sparsity, here $\beta_{ql}^{(t)}$ are sparsity parameters in [0,1], with $\beta_{ql}^{(t)} \equiv 1$ corresponding to the particular case of a complete weighted graph. More details can be found in Ambroise and Matias (2012). Lastly, we note that the VEM algorithm has been proposed to infer the model's parameters and cluster nodes.

2.5 Third-party models

In the previous sections, we defined several statistical models aimed at modeling clusters in networks. We now consider several statistical models and methods that we latter use in the Chapters 4 and 5 to derive new methodologies for network analysis. First the state space model is presented to model temporal data. Then we derive the latent Dirichlet allocation model for text analysis.

2.5.1 State space model

The state space model (SSM) (Bishop, 2006; Minka, 1999) is a general model which relies on latent variables to model sequential data. When the latent variables are discrete, the SMM model corresponds to the HMM model. In the case of Gaussian latent variables, it is usually described as a linear dynamical system.

First, consider a sequence of N observations denoted by X=$\{x_1, \ldots, x_N\}$. Each observation x_n is associated with a latent variable Z_n such that the sequence $\{Z_1, \ldots, Z_N\}$ of variables is assumed to follow a first-order Markov chain, as presented in Figure 2.17. In this model, the distribution $p(Z_n|Z_{n-1})$ of Z_n is conditioned on the value of the previous observation Z_{n-1}. That is to say, Z_n is independent of all previous variables except the most recent one Z_{n-1}. Therefore,

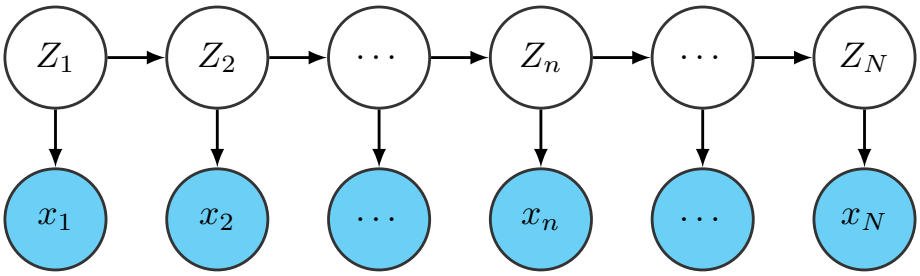


Figure 2.18: Graphical model for state space model represents sequential data using a Markov chain of latent variables.

the corresponding joint distribution is given by:

$$\begin{aligned} p(Z_1, \ldots, Z_N) &= \prod_{n=1}^{N} p(Z_n|Z_1, \ldots, Z_{n-1}) \\ &= p(Z_1) \prod_{n=2}^{N} p(Z_n|Z_{n-1}). \end{aligned} \tag{2.3}$$

The key property of the SSM model, as described by the graphical model presented in Figure 2.18, is that the observations x_{n-1} and x_n are independent given Z_n. Applied to the complete sequence of observations, this assumption leads to the following joint distribution:

$$p(x_1, \ldots, x_N, Z_1, \ldots, Z_N) = p(Z_1)\Big[\prod_{n=2}^{N} p(Z_n|Z_{n-1})\Big]\Big[\prod_{n=1}^{N} p(x_n|Z_n)\Big]. \tag{2.4}$$

In the case of a SSM model, the main goal of the inference task consists in estimating each posterior distribution $\gamma(Z_n) = p(Z_n|X)$ of Z_n given all the observed data in X. As mentioned previously, if the latent variables are discrete, then the SSM model boils down to the HMM model. The distributions $\gamma(Z_n)$ can then be computed analytically using the forward-backward algorithm (Rabiner, 1989) which is defined in Appendix A.

Linear dynamical systems We now consider some models, so called linear dynamical systems, which correspond to a specific type of SSM model where the latent variables are Gaussian random variables. In the following, the dimension of the vectors Z_n is denoted by K.

First, the distribution over the sequence of latent variables $\{Z_1, \ldots, Z_N\}$ is given by:

$$\begin{aligned} p(Z_1) &= \mathcal{N}(Z_1|\mu_0, V_0), \\ p(Z_n|Z_{n-1}) &= \mathcal{N}(Z_n|AZ_{n-1}, \Gamma), \\ p(x_n|Z_n) &= \mathcal{N}(x_n|CZ_n, \Sigma). \end{aligned}$$

The matrices A and C are transition matrices while Γ, Σ, and V_0 are covariance matrices. They are all of size $K \times K$. The vector μ_0 is of dimension K. These distributions come from the linear system of equations:

$$\begin{cases} Z_1 = \mu_0 + u \\ Z_n = AZ_{n-1} + \omega \\ x_n = CZ_n + v, \end{cases}$$

where the noise terms are supposed to have a Gaussian distribution:

$$\begin{cases} \omega \sim \mathcal{N}(0, \Gamma) \\ v \sim \mathcal{N}(0, \Sigma) \\ u \sim \mathcal{N}(0, V_0). \end{cases}$$

Estimates of the model parameters $\theta = \{A, C, \Gamma, \Sigma, V_0, \mu_0\}$ are usually obtained through maximum likelihood using the expectation maximization (EM) algorithm given in Section 2.14. This algorithm requires an analytical expression of each posterior distribution $\gamma(Z_n)$, given the observed data. These distributions can be obtained using an algorithm similar to the forward-backward algorithm for the HMM model. In the case of linear dynamical systems, the forward recursions are know as the Kalman filter equations (Kalman, 1960). Furthermore, the backward recursions are related to the Kalman smoother or Rauch-Tung-Striebel (RTS) equations (Rauch et al., 1965a). Essentially, the sums involved in the update equations of the forward-backward algorithm presented in Appendix A are replaced with integrals. As for the HMM model, an exact expression can be obtained through the use of recursions for the $\gamma(Z_n)$. Denoting

$$\gamma(Z_n) = \hat{\alpha}(Z_n)\hat{\beta}(Z_n),$$

where $\hat{\alpha}(Z_n) = p(Z_n|x_1, \ldots, x_n)$ and $\hat{\beta}(Z_n) = p(x_n, \ldots, x_N|Z_n)$. Proposition 2.5.1 gives the update equations for the $\hat{\alpha}(Z_n)$ terms. The terms $\hat{\alpha}(Z_n)$ are known as the forward messages while the terms $\hat{\beta}(Z_n)$ are called backward messages.

Proposition 2.5.1 *Since the initial forward message $\hat{\alpha}(Z_1)$ is Gaussian by construction and because each of the factors is Gaussian, all subsequent messages will be Gaussian:*

$$\hat{\alpha}(Z_n) \sim \mathcal{N}(Z_n|\mu_n, V_n),$$

with

$$\mu_n = A\mu_{n-1} + K_n(x_n - CA\mu_{n-1}),$$

$$V_n = (I - K_nC)P_{n-1},$$

where $K_n = P_{n-1}C^T(CP_{n-1}C^T + \Sigma)^{-1}$ and $P_{n-1} = AV_{n-1}A^T + \Gamma$.

The proof relies on Equation (A.19) and on some formal properties of marginal and conditional Gaussian distributions. In particular, given a marginal Gaussian distribution for x and a conditional distribution for y given x in the form:

$$p(x) = \mathcal{N}(x|\mu, \Lambda^{-1}),$$

and

$$p(y|x) = \mathcal{N}(y|Ax + b, L^{-1}),$$

then the marginal distribution of y as well as the conditional distribution of x given y are:

$$p(y) = \mathcal{N}(y|A\mu + b, L^{-1} + A\Lambda^{-1}A^T),$$

and

$$p(x|y) = \mathcal{N}(x|\Sigma(A^T L(y - b) + \Lambda\mu), \Sigma),$$

where $\Sigma = (\lambda + A^T LA)^{-1}$.

So far, we have solved the inference problem of finding the posterior marginal $\hat{\alpha}(Z_n)$ for a node Z_n given all the observations from x_1 up to x_n. To complete the inference, we can formulate the backward recursion using $\gamma(Z_n)$ rather than $\beta(Z_n)$ since $\gamma(Z_n)$ is the following Gaussian distribution:

$$\gamma(Z_n) = \hat{\alpha}(Z_n)\hat{\beta}(Z_n) = \mathcal{N}(Z_n|\hat{\mu}_n\hat{V}_n). \tag{2.5}$$

The parameters $\hat{\mu}_n$ and $\hat{V}_n$ can be computed from the backward recursion equation (A.26) multiplied by $\hat{\alpha}(Z_n)$, such that:

$$\hat{\beta}(Z_n)\hat{\alpha}(Z_n) = \hat{\alpha}(Z_n) \int_{Z_{n+1}} \frac{\hat{\beta}(Z_{n+1})}{c_{n+1}} p(x_{n+1}|Z_{n+1})p(Z_{n+1}|Z_n). \tag{2.6}$$

Then, replacing $\hat{\beta}(Z_{n+1})$ by $\gamma(Z_{n+1})/\hat{\alpha}(Z_{n+1})$ according to the equation (2.5), and using equation (A.19) we find:

$$\gamma(Z_n) = \hat{\alpha}(Z_n) \int_{Z_{n+1}} \frac{\gamma(\hat{Z}_{n+1})}{\hat{\alpha}(Z_{n+1})c_{n+1}} p(x_{n+1}|Z_{n+1})p(Z_{n+1}|Z_n). \tag{2.7}$$

Finally, equation (A.19) is used to obtain the following expression for $\hat{\alpha}(Z_{n+1})c_{n+1}$:

$$\hat{\alpha}(Z_{n+1})c_{n+1} = p(x_{n+1}|Z_{n+1}) \int_{Z_n} \hat{\alpha}(Z_n)p(Z_{n+1}|Z_n), \tag{2.8}$$

where $c_n = p(x_n|x_1, \ldots, x_{n-1})$ is also a Gaussian distribution:

$$c_n = \mathcal{N}(CA\mu_{n-1}, CP_{n-1}C^T + \Sigma).$$

Furthermore, replacing (2.8) in the equation (2.7), and using the expression for $\hat{\alpha}(Z_n)$, as provided in proposition 2.5.1, we obtain:

$$\hat{\mu}_n = \mu_n + V_n A^T (P_n)^{-1}\Big(\hat{\mu}_{n+1} - A\mu_n\Big),$$
$$\hat{V}_n = V_n + V_n A^T (P_n)^{-1}\Big(\hat{V}_{n+1} - P_n\Big)V_n A^T (P_n)^{-1}.$$

SSM Example

Let us consider an example of the use of the SSM model. To this end, we use the MARSS package in R proposed by Holmes et al. (2012). This package provides maximum-likelihood parameter estimation for constrained and unconstrained linear multivariate autoregressive state-space models, for multivariate time-series data analysis. The package relies primarily on an EM algorithm.

We simulated a sequence of 10 time points where at each time n two states were considered for the latent variable Z_n. The Γ and Σ were built as diagonal matrices with respective 0.2 and 0.1 variances. Also, the matrices A, C and V_0 were set equal to the identity matrix. Finally, we fixed all the components of the vector μ_0 to zero. Therefore, the equations for this example are as follows:

1- The initial state vector is specified at $n = 0$:

$$\begin{bmatrix} Z_1^{(0)} \\ Z_2^{(0)} \end{bmatrix} \sim MVN(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}). \tag{2.9}$$

2- Equation for two-state processes:

$$\begin{bmatrix} Z_1^{(n)} \\ Z_2^{(n)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Z_1^{(n-1)} \\ Z_2^{(n-1)} \end{bmatrix} + \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}, \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \sim MVN(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}). \tag{2.10}$$

3- The multivariate observation component in this model is as follows:

$$\begin{bmatrix} x_1^{(n)} \\ x_2^{(n)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Z_1^{(n)} \\ Z_2^{(n)} \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \sim MVN(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}). \tag{2.11}$$

From the data simulated according to (2.9), (2.10) and (2.11), we used the MARSS package to predict the states of the observations, at the 10 time points. The inference also led to the estimation of the model parameters. Figure 2.19 shows the actual (solid red lines) and estimated (solid black lines) values of the states of x_n. Furthermore, their two standard errors (green dotted lines) are provided. Additionally, we give the model parameter estimates for Γ and Σ:

$$\begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}.$$

These results highlight the interest of the forward-backward and EM algorithms to provide relevant estimates of the parameters and hidden states.

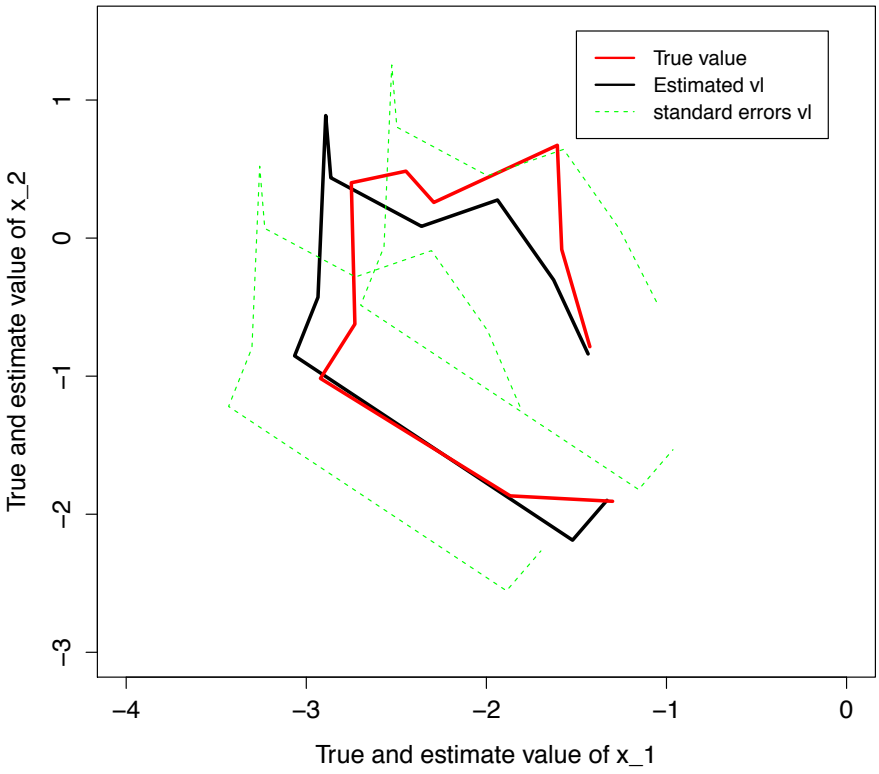


Figure 2.19: Actual (solid red lines) and estimated (solid black lines) values of the states of x_n and the 2 standard error deviations(green dotted lines).

2.5.2 Latent Dirichlet allocation model

The main statistical model for text analysis is the latent Dirichlet allocation (LDA) model (Blei et al., 2003). This model was introduced as an extension of the probabilistic latent semantic analysis (pLSI) approach of Hofmann (1999). The pLSI method characterizes each word within a document by relying on a mixture model over topics. Depending on the (unknown) topics, the words of the dictionary have various probabilities to occur. pLSI can be seen as the probabilistic generalization of the LSI methodology (Papadimitriou et al., 1998) which considers singular value decompositions to extract information from the collection of documents. Conversely, the mixture of unigrams model (Nigam et al., 2000) does not associate a topic to each word within the documents. It rather associates a unique topic to each document, depending on the words it is built on. Since the publication of the original work of Blei et al. (2003), LDA has become a standard tool in statistical text analytics and is even used in different scientific fields such as image analysis (Lazebnik et al., 2006) or transportation research (Côme et al., 2014) for instance. The idea at the core of LDA is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Note that LDA is similar to pLSI except that the topic distribution in LDA has a Dirichlet distribution.

Let us consider a corpus, i.e. a collection of D documents made out of words from a dictionary. In the following, the size of the dictionary is denoted by V. Each document $W_d = (W_{dn})_n$ is a sequence of N_d words. In terms of coding, each word W_{dn} is a vector of size V such that $(W_{dn})_j = 1$ if word n of document W_d is the vth word of the dictionary, 0 otherwise.

First, each document W_d is associated with a latent vector θ_d assumed to be drawn from a Dirichlet distribution:

$$\theta_d \sim \mathrm{Dir}\left(\alpha = (\alpha_1, \ldots, \alpha_K)\right),$$

where K is the number of topics used. Thus, θ_d is a vector of size K such that $\sum_{k=1}^{K} \theta_{dk} = 1, \forall d$. The parameter θ_{dk} is the probability to find words of topic k in the document. The set of latent vectors θ_d is denoted by $\theta = (\theta_d)_d$. Then, the nth word W_{dn} is associated with a latent topic vector Z_{dn} assumed to be drawn from a multinomial distribution:

$$Z_{dn} \sim \mathcal{M}\left(1, \theta_d\right). \tag{2.12}$$

By construction $\sum_{k=1}^{K} (Z_{dn})_k = 1, \forall d$. If $(Z_{dn})_k = 1$ then word n of document W_d is from topic k. Finally, given Z_{dn}, the word W_{dn} is supposed to be drawn from a multinomial distribution

$$W_n^d | (Z_{dn})_k = 1 \sim \mathcal{M}\left(1, \beta_k = (\beta_{k1}, \ldots, \beta_{kV})\right), \tag{2.13}$$

where β_k is a vector of size V such that $\sum_{v=1}^{V} \beta_{kv} = 1, \forall k$. The parameter β_{kv} is the probability for word v of the dictionary to appear in topic k. The set of all vectors β_k is denoted by $\beta = (\beta_{kv})_{kv}$.

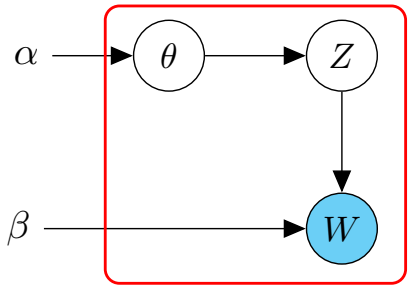


Figure 2.20: Graphical representation of the LDA model.

All the latent variables Z_{dn} and θ_d are assumed to be sampled independently and, given the latent variables, the words W_{dn} are assumed to be independent. Denoting $W = (W_n^d)_d$, this leads to the following joint distribution and the corresponding LDA graphical model is provided in Figure 2.20:

$$p(W, Z, \theta|\beta) = p(W|Z, \beta)p(Z|\theta)p(\theta|\alpha),$$

where

$$\begin{aligned} p(W|A, Z, \beta) &= \prod_{d=1}^{D} \prod_{n=1}^{N_d} p(W_n^d|Z_{dn}, \beta) \\ &= \prod_{d=1}^{D} \prod_{n=1}^{N_d} \prod_{k=1}^{K} p(W_n^d|\beta_k)^{Z_{dn}}, \end{aligned}$$

and

$$p(Z|\theta) = \prod_{d=1}^{D} \prod_{n=1}^{N_d} p(Z_{dn}|\theta_d).$$

Furthermore

$$p(\theta) = \prod_{d=1}^{D} \mathrm{Dir}(\theta_d; \alpha).$$

Many inference procedures have been proposed in the literature ranging from VEM (Blei et al., 2003) to collapsed VBEM (Teh et al., 2006) as well as Gibbs sampling (Newman et al., 2007). We note that a limitation of LDA would be the inability to take into account possible topic correlations. This is due to the use of the Dirichlet distribution to model the variability among the topic proportions. To overcome this limitation, the correlated topic model (CTM) was also developed by Lafferty and Blei (2006). Similarly, the relational topic model (RTM) (Chang and Blei, 2009) models the links between documents as binary

random variables conditioned on their contents, but ignoring the community ties between the authors of these documents. Notice that the "itopic" model ((Sun et al., 2009)) extends RTM to weighted networks.

LDA example

In this section, we now illustrate the use of the LDA model and the corresponding methodology for document analysis. To this end, we consider two documents related to medicine and politics in the UK. The corpus of this two documents is analyzed using the topicmodels R package (Grun and Hornik, 2013) which implements various inference algorithms for LDA. We used the VEM algorithm with $K = 2$. In the following, we give samples of the documents with some words indicated in red. They correspond to words associated to topics discovered by the method. Thus, in the first document, the words in red are clearly associated to cancer research while they are related to parliamentary issues in the second.

"Cancer *is a group of* diseases *involving* abnormal *cell growth with the potential to invade or spread to other parts of the* body. *Not all* tumors *are* cancerous*; benign tumors do not spread to other parts of the body. Possible signs and symptoms include: a new lump, abnormal bleeding, a prolonged cough, unexplained weight loss, and a change in bowel movements among others. While these symptoms may indicate cancer, they may also occur due to other issues.There are over 100 different known cancers that affect humans...*"

Second one, take the subject about the politics such as:

"*The* political *future of* the United Kingdom *has become clearer after the* results *of the general election emerged around the country.* David Cameron *says he hopes to* govern *for all of the UK after the Conservatives took 331* seats *- enough to form a slender majority in the Commons. Labour has been all but wiped out by the SNP in Scotland and suffered a disappointing set of results elsewhere, while the Lib Dems are left with just eight MPs after many party heavyweights such as Vince* Cable *and Danny* Alexander *lost their seats..*"

We show in Table 2.1 the six words mainly associated with each topic. Again, the topics retrieved correspond to the subjects of the documents.

Second, we give the results for θ. Clearly, document 1 is made out of words from topic 2. Conversely, document 2 has words from topic 1.

These results illustrate that LDA is a flexible model which can be used within an inference framework to provide relevant results when analyzing a corpus of documents.

Topic 1	Topic 2
"results"	"abnormal"
"seats"	"body"
"alexander"	"cancer"
"cameron"	"symptoms"
"political"	"parts"
"commons"	"affect"
....	....
....	....

Table 2.1: The main six words associated with each topic, as found by the LDA methodology.

	Topic 1	Topic 2
doc 1	0.0004662839	0.9995337161
doc 2	0.9995147025	0.0004852975

Table 2.2: Matrix of topic proportions for each document.

2.6 Inference algorithms and model selection

This section introduces several variational techniques for probabilistic models. Some of them are used in Chapters 4 and 5 to extract information from networks. We also derive some model selection criteria to estimate the number of clusters from the data.

2.6.1 Variational inference algorithms

In machine learning, to obtain estimates of mixture model parameters, the standard approach is the expectation maximization (EM) algorithm, which was originally developed to find the maximum likelihood for a model, and also to find maximum a posterior (MAP) estimates. Here, we present the variational EM (VEM) (Neal and Hinton, 1998; Jordan et al., 1999; Hathaway, 1986) and variational Bayes EM (VBEM) (Corduneanu and Bishop, 2001; Svensén and Bishop, 2005) algorithms, which are extensions of the EM algorithm, in the variational framework. To this end, we first pause to explain the main idea and notation for the EM algorithm, so that it will be easier to explain the VEM and VBEM algorithms.

We note that in this section, we will deal with the variational framework, which derives from the EM algorithm, but remember that these are not the only methods which could be used to do inference for mixture models. Gibbs sampling, proposed by Geman and Geman (1984); Rabiner (1989); Gelfand and Smith (1990), is an algorithm based on Monte Carlo Markov Chain (MCMC) techniques, is one possible way to perform inference for such models.

Expectation maximization algorithm The expectation maximization algorithm, or EM, is a general technique for finding (locally) maximum likelihood parameters of a statistical model with missing data. This algorithm is based on a recursive function where at each iteration, there are two steps: the Expectation or E-step, and the Maximization or M-step. The algorithm was first presented in Dempster et al. (1977) and Krishnan and McLachlan (1997). The basic idea of the EM algorithm is to associate with the given incomplete-data problem, complete-data, where the problem's maximum likelihood (ML) estimation is computationally more tractable.

Consider a mixture model that has a set of parameters denoted θ, and all of the observations are drawn from this model, denoted X, and all of the hidden variables by Z, where Z is a discrete variable. The goal is to maximize the likelihood function $p(X|\theta)$. The direct optimization of $p(X|\theta)$ is difficult, and in cases when we introduce the latent variable Z, in the likelihood function we obtain:

$$p(X|\theta) = \sum_{Z} p(X, Z|\theta).$$

This summation involves Q^N terms if there are N observations and Q clusters, and quickly becomes intractable. To tackle such a problem, we can apply the EM algorithm, which has had great success on a large variety of mixture models. We note that $p(X, Z|\theta)$ is called the complete-data likelihood.

Proposition 2.6.1 *Given a distribution $q(Z)$ defined over latent variables, and for any choice of $q(Z)$, we have the following decomposition:*

$$\log p(X|\theta) = \mathcal{L}(q, \theta) + KL(q(.) \parallel p(.|X, \theta), \tag{2.14}$$

where $\mathcal{L}$ is defined as follows:

$$\mathcal{L}(q, \theta) = \sum_{z} q(Z) \log \frac{p(X, Z|\theta)}{q(Z)},$$

and KL denotes the Kullback-Leibler divergence between the true and approximate posterior distributions:

$$KL(q(.) \parallel p(.|X, \theta)) = -\sum_{z} q(Z) \log \frac{p(Z|X, \theta)}{q(Z)}.$$

Proof of Prop. 2.6.1.

$$\begin{aligned}
\mathcal{L}(q,\theta) + KL(q(.) \parallel p(.|X,\theta) &= \sum_z q(Z) \log \frac{p(X,Z|\theta)}{q(Z)} - \sum_Z q(Z) \log \frac{p(Z|X,\theta)}{q(Z)} \\
&= \sum_Z q(Z) \log p(X,Z|\theta) - \sum_Z q(Z) \log p(Z|X,\theta) \\
&= \sum_Z q(Z) \log \frac{p(X,Z|\theta)}{p(Z|X,\theta)} \\
&= \sum_Z q(Z) \log \frac{p(Z|X,\theta)p(X|\theta)}{p(Z|X,\theta)} \\
&= \log p(X|\theta) \sum_Z q(Z) \\
&= \log p(X|\theta). \quad \blacksquare
\end{aligned}$$

Note that $\mathcal{L}(q,\theta)$ is a functional of the distribution $q(Z)$ and a function of θ. Moreover, the Kullback-Leibler satisfies:

$$KL(q(.) \parallel p(.|X,\theta)) \geqslant 0,$$

and null if there is equality between $q(Z)$ and $p(Z|X,\theta)$. In this case, $\mathcal{L}(q,\theta)$ is the lower bound of $\log p(X|\theta)$:

$$\mathcal{L}(q,\theta) \leq \log p(X|\theta).$$

Looking for the best approximation of the posterior distribution $p(Z|\theta)$ in the KL divergence sense becomes equivalent to searching for a distribution $q(\cdot)$ that maximizes the lower bound $\mathcal{L}$ of the integrated log-likelihood.

The EM algorithm functions recursively, in order to maximize the log-likelihood, which is not dependent on the distribution of Z. This maximization is done when the KL divergence is zero. In this case, $q(Z)$ equals the posterior distribution of Z, defined by $p(Z|X,\theta)$, which leads to the lower bound being equal to the log-likelihood. Therefore, in this step, the algorithm computes $p(Z|X,\theta_{old})$, and the lower bound takes the form:

$$\begin{aligned}
\mathcal{L}(q,\theta) &= \sum_Z p(Z|X,\theta_{old}); \log p(X,Z|\theta) - \sum_Z p(Z|X,\theta_{old}) \log p(Z|X,\theta_{old}) \\
&= \mathcal{Q}(\theta,\theta_{old}) + const.
\end{aligned}$$

During the M step, the distribution $q(Z)$ is fixed, and the lower bound $\mathcal{L}(q,\theta)$ is maximized with respect to θ, to give a new value θ_{new}. Each recursion consists of an E-step, when the value of θ is fixed and noted θ_{old}, and we maximize the lower bound with respect to $q(Z)$. These two steps are repeated until convergence is obtained.

Thus, we can summarize the EM algorithm as follows:

- First, we initialize the parameters θ_{old} to some random values or from an arbitrary algorithm.
- Second, during the E-step, we compute $p(Z|X, \theta_{old})$.
- Then, we maximize $\mathcal{Q}(\theta, \theta_{old})$ with respect to θ. We obtain θ_{new}.
- The E and M-steps are repeated until convergence.

Since the EM algorithm defined by Dempster et al. (1977), many extensions have been created to find solutions for diverse problems in estimation, and to find node clusters in data sets. One example is classification EM (CEM) (Celeux and Govaert, 1991), which maximizes a classifier's likelihood and accelerates convergence of the algorithm. CEM is obtained from the classical EM algorithm by adding a classification C-step. This C-step, in each iteration, places each observation in one class according to the MAP rule. This classification approach gives a biased and not consistent estimation of θ, and from a theoretical point of view, it is preferable to use a mixture approach and the EM algorithm. Nevertheless, CEM convergence is much faster than EM, and may be useful when we have time constraints or a large data set.

Another example is the variational EM (VEM) algorithm, which we later define in Section 2.6.1, and use in Chapters 3 and 4 to obtain an approximation of the posterior distribution over the model parameters. As already mentioned, there is also variational Bayes EM (VBEM), defined in Section 2.6.1, which is an approach to approximate the full posterior distribution of the model parameters and latent variables, given the observed data X.

Variational EM algorithm

Above, we have defined one among many methods that can be used to do inference of mixture models with i.i.d. latent variables, but this situation does not always hold. For instance, in the SBM case, the latent variables are not independent. Below, we will present variational expectation maximization (VEM) methods as described by Jordan et al. (1999) and in the tutorial by Jaakkola (2001).

The VEM algorithm also consists of two alternating steps, which focus on approximating the local probability distribution at the nodes of a graphical model. In the variational E-step, we suppose that the current values of the model parameters are $\theta_{old} = (\alpha_{old}, \Pi_{old})$, and that the lower bound $\mathcal{L}(q, \theta_{old})$, which is defined in Section 2.6.1, is maximized with respect to $q(Z)$, while the model parameters are fixed. In this step, we put a conditions of the distribution of $q(Z)$ in order to obtain a tractable algorithm; for example, that $q(Z)$ can be factorized as follows:

$$q(Z) = \prod_{i=1}^{N} q(Z_i).$$

Then, optimization (variational M step) for the model parameters occurs for a fixed distribution $q(Z)$, to give $\theta_{new} = (\alpha_{new}, \Pi_{new})$. These two steps are repeated until convergence.

We note that Daudin et al. (2008) used the VEM algorithm (Algorithm 3 here) to jointly estimate SBM model parameters and cluster the vertices of a network, where $\theta = (\alpha, \Pi)$.

Algorithm 3: The variational EM algorithm in case where θ is the model parameters of the SBM.

INITIALIZATION

Initialization of $\theta^0 = (\alpha^0, \Pi^0)$
$\alpha_q^{new} \leftarrow \alpha_q^0 \ \forall\, q$
$\Pi_q^{new} \leftarrow \Pi_q^0 \ \forall\, q$
OPTIMIZATION

repeat

$\alpha_q^{old} \leftarrow \alpha_q^{new} \ \forall\, q$
$\Pi_q^{old} \leftarrow \Pi_q^{new} \ \forall\, q$
Variational E-step

update the variational variable of $q(Z)$ by finding $\text{argmax}_{q(z)}\mathcal{L}(q; \theta^{old})$

Variational M-step

calculate $\alpha_q^{new} = \text{argmax}_\alpha \mathcal{L}(q; \theta)$

calculate $\Pi_{qr}^{new} = \text{argmax}_\Pi \mathcal{L}(q; \theta)$
until convergence of θ

Variational Bayes EM algorithm

In the previous sections, we have defined the main idea of the EM algorithm, and its extensions in the variational framework, to estimate mixture model parameters. Here, we shall now present a new approach for estimating the marginal likelihood of probabilistic models with latent variables or incomplete data (Corduneanu and Bishop, 2001; Svensén and Bishop, 2005). This method constructs and optimizes a lower bound on the marginal likelihood using variational calculus, resulting in an iterative algorithm generalizing EM, by maintaining a posterior distribution over both the latent variables and parameters.

Let us denote X an observed data set, Z the corresponding classification matrix, and θ the parameters. In the variational Bayesian framework, all model parameters are considered as random variables drawn from a prior distribution $p(\theta)$. The goal here is to approximate the full distribution $\text{p}(Z, \theta|X)$. Thus,

the log marginal likelihood, or integrated observed data log-likelihood as in Equation 2.14, can be decomposed into the following terms:

$$\log p(X) = \mathcal{L}(q(.)) + KL(q(.) \parallel p(.|X)),$$

where

$$\begin{aligned}\log p(X) &= \log\Big(\int p(X,\theta)d\theta\Big) \\ &= \log\Big(\sum_{Z}\int p(X,Z,\theta)d\theta\Big).\end{aligned}$$

The lower bound of the marginal log likelihood is the functional $\mathcal{L}$ such that:

$$\mathcal{L}(q) = \sum_{z}\int q(Z,\theta)\log\frac{p(X,Z,\theta)}{q(Z,\theta)}d\theta,$$

and

$$KL(q(.) \parallel p(.|X)) = -\sum_{z}\int q(Z,\theta)\log\frac{p(Z,\theta|X)}{q(Z,\theta)}d(\theta).$$

Here, for any choice of distribution over latent variables and parameters $q(Z,\theta)$, maximizing the lower bound $\mathcal{L}(q)$ over these distributions is equivalent to minimizing the KL-divergence between $q(Z,\theta)$ and $p(Z,\theta|X)$. Therefore, we look for an approximation $q(Z,\theta)$ of $p(Z,\theta|X)$, since the model parameters are random variables. To obtain a tractable algorithm, we assume that the distribution over the latent variable and parameters can be factorized:

$$q(Z,\theta) = q(\theta)q(Z) = q(\theta)\prod_{i=1}^{N} q(Z_i).$$

Then, during the variational Bayes E-step, the lower bound is maximized with respect to the distribution $q(Z)$, whereas the distribution of $q(\theta)$ in this step is fixed and takes the initials values. During the variational Bayes M-step, the approximation $q(Z)$ is fixed and used to compute the lower bound $\mathcal{L}(q)$, which is then maximized with respect to $q(\theta)$.

Remark. In all definitions in this section, we consider that the latent variables Z are discrete. However, these can be replaced by continuous variables, and in this case, sums should be replaced by integrals.

2.6.2 Model selection criteria

So far, we have defined all models with the number Q of classes fixed, which is not necessarily the case in real data, where we also should try to find this number. Here, we describe some model selection criteria existing in the literature which estimate the number of Q from the data. In the context of model selection, we

assume that there are data and a set of models (set of values of Q). Classically, it is assumed that there is a single correct or, at least, best model, so the goal is to select Q^* such that a given criterion is maximized. Although the parameter of that model is unknown, it is assumed that it can be estimated. Therefore, classical inference (mentioned in Section 2.6.1) is often involve to estimate parameters for each value of Q in the set.

Akaike's information criterion

The Akaike Information Criterion (AIC) Akaike (1973) is a way of selecting a model from a set of models. The chosen model is the one that minimizes the KL distance between the model and the truth. This framework is based on information theory.

Let X be the data set we are modeling, which is supposed drawn from a mixture model with parameters θ and a fixed number Q of classes. $\mathcal{D}$ is a set of models of the same size as X, with different values of Q, which are supposed candidate parametric models. Here, the goal is to select one model among $\mathcal{D}$ for which the corresponding information loss is minimal. Suppose that Y is a model belonging to $\mathcal{D}$, and $\hat{\theta}$ is an estimate of θ using the EM algorithm. Since the EM algorithm converges to a local rather than necessarily the global one, the distribution $p(Y|\hat{\theta})$ can be seen as an approximation of the true distribution $p(Y)$ which generated X. Model selection can be approached in terms of the KL divergence between $p(Y)$ and $p(Y|\hat{\theta})$:

$$KL\Big(p(.) \parallel p(.|\hat{\theta})\Big) = -\int p(Y) \log \frac{p(Y|\hat{\theta})}{p(Y)} dY \tag{2.15}$$

$$= \int p(Y) \log p(Y) dY - \int p(Y) \log p(Y|\hat{\theta}) dY. \tag{2.16}$$

From equation (2.16), we can see that only the term on the right depends on the fitted model $p(Y|\hat{\theta})$, which can be rewriten using the expectation according to $p(Y)$ as follows:

$$f(Y) = \int p(Y) \log p(Y|\hat{\theta}) dY. \tag{2.17}$$

$$= E_Y[\log p(Y|\hat{\theta})]. \tag{2.18}$$

Then, as (2.18) is intractable, because $p(Y)$ is unknown, we take the expectation of $f(Y)$ over every possible data set X, and obtain:

$$E_X[f(Y)] = E_{X,Y}[\log p(Y|\hat{\theta})], \tag{2.19}$$

which can be estimated as in Peel and McLachlan (2000). In practice, only a single data set X is given, and Akaike (1973, 1974) showed that (2.19) is asymptotically equal to:

$$\log p(X|\hat{\theta}) - K, \tag{2.20}$$

where K is the total number of parameters in the model. The approximation (2.20) corresponds to the AIC.

2.6.3 Bayesian information criterion

Suppose that we have a model with N observations, given in a matrix X, with θ a set of parameters, including κ, the total number of parameters in the model. The Bayesian information criterion (BIC) seeks to select a model M_i from a finite set $\mathcal{D} = \{M_1, \cdots, M_m\}$, by maximizing the posterior probability $p(M_i|X)$:

$$M_{BIC} = argmax_{M_i} P(M_i|X).$$

The BIC criterion relies on an asymptotic approximation of the marginal log-likelihood $\log p(X)$, given by:

$$\log p(X) \approx \log p(X|\hat{\theta}) - \frac{\kappa}{2} N, \tag{2.21}$$

where $\hat{\theta}$ is the estimation of θ using the EM algorithm. To show that this approximation, made by (Schwarz et al., 1978), is valid, we begin by writing the marginal log-likelihood according to the prior distribution over the mixture model parameters:

$$\log p(X) = \log\{\int p(X, \theta) d\theta\} = \log\{\int p(X|\theta) p(\theta) d\theta\}. \tag{2.22}$$

To show that this approximation, made by (Schwarz et al., 1978), is valid, we begin by writing the marginal log-likelihood according to the prior distribution over the mixture model parameters:

$$\log p(X) = \log\{\int p(X, \theta) d\theta\} = \log\{\int p(X|\theta) p(\theta) d\theta\}. \tag{2.23}$$

Since the integration, over all possible values of θ, is generally not tractable in (2.23), this give rise to the use of an approximation, here a second-order Taylor series, to approximate the integral, where at a point $\theta = \hat{\theta}$, we have:

$$\log p(X, \theta) \approx \log p(X, \hat{\theta}) + \nabla_{\theta=\hat{\theta}} \log p(X, \theta)^{\intercal} (\theta - \hat{\theta}) - \frac{1}{2} (\theta - \hat{\theta})^{\intercal} \mathbf{H} (\theta - \hat{\theta}), \tag{2.24}$$

where $\mathbf{H}$ is the negative Hessian matrix of $\log p(X, \theta)$ at $\hat{\theta}$.

Note that $\log p(X)$ does not depend on θ, and $\hat{\theta}$ maximizes $\log p(\theta|X)$, so for $\theta = \hat{\theta}$, and using the decomposition $\log p(X, \theta) = \log p(\theta|X) p(X)$, we have that $\nabla_{\theta=\hat{\theta}} \log p(X, \theta) = 0$. At this point we have:

$$\log p(X, \theta) \approx \log p(X, \hat{\theta}) + \log(\hat{\theta}) - \frac{1}{2} (\theta - \hat{\theta})^{\intercal} \mathbf{H} (\theta - \hat{\theta}). \tag{2.25}$$

Then, applying the exponential function to (2.25) leads to the appearance of a Gaussian distribution with mean vector $\hat{\theta}$ and covariance matrix H^{-1}. The integral of this distribution gives:

$$\int \exp\Big(-\frac{1}{2}(\theta-\hat{\theta})\mathbf{H}(\theta-\hat{\theta})\Big)d\theta = (2\Pi)^{d/2}|\mathbf{H}|^{-\frac{1}{2}}. \tag{2.26}$$

Therefore, using (2.23), (2.24) and (2.26) gives the following approximation to $p(X)$,

$$\log p(X) \approx \log\Big(\int p(X,\hat{\theta})\exp\Big(-\frac{1}{2}(\theta-\hat{\theta})^{\intercal}\mathbf{H}(\theta-\hat{\theta})\Big)d\theta\Big) \tag{2.27}$$

$$\approx \underbrace{\log p(X|\hat{\theta})}_{\text{log-likelihood}} + \underbrace{\log p(\hat{\theta}) + \frac{d}{2}\log(2\Pi) - \frac{1}{2}\log|\mathbf{H}|}_{\text{Occam Factor}}. \tag{2.28}$$

In this approximation, the first term of (2.28) represents the log-likelihood and the second part the Occam factor, which penalizes model complexity and comprises three terms. Lastly, to recover the final form of the approximation given in 2.21, we approximate $\mathbf{H}$ by the expected Fisher information matrix of the observed data (Kass and Raftery, 1995), noted $\mathbf{J}$, such as:

$$|\mathbf{J} = O(N^K)|,$$

then, we assume that the effect of the prior $\log p(\hat{\theta})$ and the terms in $O(1)$ can be ignored in (2.28).

We note that for N $>$ 7, the BIC criterion penalize more heavily than AIC criterion (2.6.2). Many experiments on real data have been done in order to assess the performance of BIC (Fienberg and Wasserman, 1981a; Dasgupta and Raftery, 1998), and generally confirm that it reduces the tendency of AIC to fit too many components.

Integrated classification likelihood criterion

An even more drastic criterion to select a relevant number of classes, called the Integrated Completed Likelihood (ICL) criterion, was proposed by Biernacki et al. (2000b). This criterion relies on an asymptotic approximation of the integrated completed data log-likelihood $\log p(X, Z)$.

Let us denote that X is a data set drawn from a mixture model with a fixed number of clusters Q, and $\theta = \{\theta_1, \theta_2\}$ is the model parameter, where θ_1 the parameter of X and θ_2 the parameter of Z. The joint probability of X and Z is given by:

$$\log p(X, Z) = \log p(X|Z) + \log p(Z). \tag{2.29}$$

To prove (2.29), we suppose that the prior $p(\theta)$ factorizes as $p(\theta) = p(\theta_1)p(\theta_2)$.

Therefore,

$$\begin{aligned}\log p(X,Z) &= \int \log p(X,Z|\theta)p(\theta)d\theta \\ &= \int \log p(X|Z,\theta)p(Z|\theta)p(\theta)d\theta \\ &= \int \log p(X|Z,\theta_1)p(Z|\theta_2)p(\theta_1)p(\theta_2)d\theta_1 d\theta_2 \\ &= \Big(\int \log p(X|Z,\theta_1)p(\theta_1)d\theta_1\Big)\Big(\int p(Z|\theta_2)p(\theta_2)d\theta_2\Big).\end{aligned}$$

Then, to approximate $\log p(X,Z)$, we apply the BIC approximation on the first term of the right hand side of 2.29, and obtain:

$$\log p(X|Z) \approx p(X|Z,\hat{\theta}_1) - \frac{M_1}{2}\log N, \tag{2.30}$$

where M_1 is the number of parameters in θ_1, whose estimator is $\hat{\theta}_1$, which maximizes the likelihood $p(X|Z,\hat{\theta}_1)$. Also, for the second part of (2.29), we use the analytical expression of (Biernacki et al., 2000b) to approximate $p(Z)$ (for more details see (Latouche, 2011)), giving:

$$\log p(Z) \approx \log p(Z|\hat{\theta}_2) - \frac{Q-1}{2}\log N, \tag{2.31}$$

where $\hat{\theta}_2 = argmax_\theta \log p(Z|\theta)$. Lastly, to approximate the complete data log-likelihood $\log p(X,Z)$, we use (2.29), (2.30) and (2.31) to give the following result:

$$\begin{aligned}\log p(X,Z) &\approx p(X|Z,\hat{\theta}_1) - \frac{M_1}{2}\log N + \log p(Z|\hat{\theta}_2) - \frac{Q-1}{2}\log N \\ &\approx p(X,Z|\hat{\theta}) - \frac{K_1+Q-1}{2}\log N.\end{aligned}$$

2.7 Conclusion

Several methods introduced in this chapter will serve as the basis for new models defined in the following chapters. In summary, we have reviewed some mixture models for static and dynamic networks, characterizing different types of edges. We have also focused on several variational techniques to perform inference, and on several criteria to select the number of clusters, when this is not fixed.

CHAPTER 3

DYNAMIC RANDOM SUBGRAPH MODEL

Contents

In recent years, many clustering methods have been proposed to extract information from networks. The principle is to look for groups of vertices with homogenous connection profiles. Most of these techniques are suitable for

static networks, that is to say, not taking into account the temporal dimension. This work is motivated by the need of analyzing evolving networks where a decomposition of the networks into subgraphs is given. Therefore, in this paper, we consider the random subgraph model (RSM) which was proposed recently to model networks through latent clusters built within known partitions. Using a state space model to characterize the cluster proportions, RSM is then extended in order to deal with dynamic networks. We call the latter the dynamic random subgraph model (dRSM). A variational expectation maximization (VEM) algorithm is proposed to perform inference. We show that the variational approximations lead to an update step which involves a new state space model from which the parameters along with the hidden states can be estimated using the standard Kalman filter and Rauch-Tung-Striebel (RTS) smoother. Simulated data sets are considered to assess the proposed methodology.

3.1 Introduction

Network analysis has become a mature discipline, since the original work of Moreno (1934), which is no longer limited to sociology and is now applied in many areas such as biology (Albert and Barabási, 2002; Barabási and Oltvai, 2004; Palla et al., 2005), geography (Ducruet, 2013) or history (Rossi et al., 2014). The growing interest in network analysis is explained partly by the strong presence of this type of data in the digital world, and by recent advances in the modeling and the processing of these data. The clustering methods allow in particular clusters of vertices sharing homogeneous connection profiles to be uncovered. Most methods look for specific structures, so called communities, which exhibit a transitivity property such that nodes of the same community are more likely to be connected (Hofman and Wiggins, 2008). A popular approach for community discovering, though asymptotically biased (Bickel and Chen, 2009), is based on the modularity score given by Girvan and Newman (2002). Alternative methods usually rely on the latent position cluster model (LPCM) of Handcock et al. (2007) which assumes that the links between the vertices depend on their positions in a social latent space.

The stochastic block model (SBM) (Wang and Wong, 1987; Nowicki and Snijders, 2001) is a flexible random graph model which can also characterize communities, but not only. It is based on a probabilistic generalization of the method applied by White et al. (1976) on Sampson's famous monastery (Fienberg and Wasserman, 1981b). The SBM model assumes that each vertex belongs to a latent group, and that the probability of connection between a pair of vertices depends exclusively on their group. Because no assumption is made on the connection probabilities, various types of structures of vertices can be taken into account. While SBM was originally developed to analyze mainly binary networks, many extensions have been proposed since to deal for instance with valued edges (Mariadassou et al., 2010) or to take into account prior information (Zanghi et al., 2010; Matias and Robin, 2014). In particular, the random subgraph model (RSM) of Jernite et al. (2014) aims at modeling categorical

edges using prior knowledge of a partition of the network into subgraphs. These known subgraphs are assumed to be made of latent clusters which have to be inferred. The vertices are then connected with a probability depending only on the subgraphs whereas the edge type is assumed to be sampled conditionally on the latent groups. This model was applied in the original paper to analyze a historical network in merovingian Gaul. Note that other extensions of SBM have focused on looking for overlapping clusters (Airoldi et al., 2008; Latouche et al., 2011). The inference of SBM like models is usually done using variational expectation maximization (VEM) (Daudin et al., 2008), variational Bayes EM (VBEM) (Latouche et al., 2012), or Gibbs sampling (Nowicki and Snijders, 2001). Moreover, we emphasize that various strategies have been derived to estimates the number of corresponding clusters using model selection criteria (Daudin et al., 2008; Latouche et al., 2012), allocation sampler (Mc Daid et al., 2013), greedy search (Côme and Latouche, 2015), or non parametric schemes (Kemp et al., 2006).

Recently, a few attempts have been made to extend the models mentioned previously in order to deal with dynamic networks. The main idea consists in introducing temporal processes in order to characterize the temporal evolution of nodes and edges through time. Thus, Yang et al. (2011) proposed a dynamic version of SBM allowing a node to switch its class at time $t + 1$ depending on its state at time t. The switching probabilities are all characterized by a transition matrix. The alternative extension for SBM of Xu and Hero III (2013) focuses on modeling the temporal changes through a state space model and relies on the Kalman filter for inference. Contrary to Yang et al. (2011); Xu and Hero III (2013) treated the edge probabilities as time varying parameters. In parallel, the mixed membership SBM (MMSBM) of Airoldi et al. (2008), capable of characterizing overlapping clusters, was adapted to deal with dynamic networks by Xing et al. (2010); Ho et al. (2011) and Kim and Leskovec (2013a). Moreover, Sarkar and Moore (2005) derived a dynamic version of the LPCM model of Handcock et al. (2007) keeping the transitivity property that nodes which are close in a social latent space should be more likely to connect. Finally, we would like to highlight the work of Dubois et al. (2013) and Heaukulani and Ghahramani (2013). In (Dubois et al., 2013) a non homogeneous Poisson process is considered. Thus, contrary to most clustering models for dynamic networks, a continuous time period is taken into account and events, *i.e.* the creation or removal of an edge, occur one at a time. While models usually focus on modeling the dynamic of networks through the evolution of their latent structures, Heaukulani and Ghahramani (2013) extended the dynamic latent feature model of Foulds et al. (2011) to define how observed social interactions can affect future unobserved latent structures. In the same vein, a dynamic model inspired by SBM was proposed recently by Xu (2015).

In this chapter, we aim at modeling dynamic networks with binary or more generally typed edges, for which a partition of the nodes is given. As an example, we will consider an original network, built from printed Lloyd's voyage records and describing maritime flows between ports where the geographical positions of the ports play an important role. The partition was obtained by associating

each port to a region according to its geographical position. Figure 3.1 presents the evolution of network navigations, for 23 years between October 1985 and October 2008. A (given) partition of the nodes is seen here as a decomposition of the network into known subgraphs that we propose to model using unobserved clusters that have to be inferred from the data in practice. Thus, considering a slightly different version of the original RSM model of Jernite et al. (2014) and relying on a state space model as in (Xing et al., 2010), we propose a new random graph model for evolving networks that we call the dynamic RSM (dRSM) model. The model focuses on describing the network dynamic by characterizing the evolution of the cluster proportions within the known subgraphs. A logistic transformation is used to link the hidden states and the clusters proportions, as in (Ahmed and Xing, 2007; Blei and Lafferty, 2007b). The inference of the model is done using a VEM algorithm.

The organization of this chapter as follows. In Section 3.2, we introduce the dRSM model along with an inference procedure in Section 3.3. Variational techniques are considered and a model selection criterion is derived. Finally, the methodology is tested on simulated data in Section 3.4.

3.2 The dynamic random subgraph model

This section presents the context of the work and introduces the dRSM model along with the modeling of its dynamic. The joint distribution associated with the model is also detailed.

3.2.1 Context and notations

We consider a set of T networks $\{\mathcal{G}^{(t)}\}_{t=1}^{T}$, where $\mathcal{G}^{(t)}$ is a directed graph observed at time t. Each $\mathcal{G}^{(t)}$ is represented by its $N \times N$ adjacency matrix $X^{(t)}$ where N denotes the number of nodes. The edge $X_{i,j}^{(t)}$, describing the relationship between nodes i and j, is assumed to take its values in $\{0, \dots C\}$ such that $X_{ij}^{(t)} = c$ means that nodes i and j are linked by a relationship of type c at time t and $X_{ij}^{(t)} = 0$ indicates the absence of relationship between the two nodes at time t. Note that no self loops are considered, *i.e.* the connection of a node to itself, thus $X_{ii}^{(t)} = 0$, $\forall\, i, t$.

Moreover, a partition $\mathcal{P}$ of the network into S classes of vertices is assumed to be given. We emphasize that the observed partition induces a decomposition of the graph into subgraphs where each class of vertices corresponds to a specific subgraph. To describe the subgraph membership of each vertex, the variable s is introduced. The variable takes its values in $\{1, \dots S\}$ and is such that s_i indicates the subgraph of vertex i. In some cases, and in order to clarify the equations, we will also consider the indicator variables y_{is} such that $y_{is} = 1$ if node i is in subgraph s, 0 otherwise. Finally, because the vertex i can only belong to a single subgraph, we have $\sum_{s=1}^{S} y_{is} = 1$.

Our goal is to cluster at each time t the N nodes into Q latent groups with homogeneous connection profiles, *i.e.* find an estimate of the set Z of latent

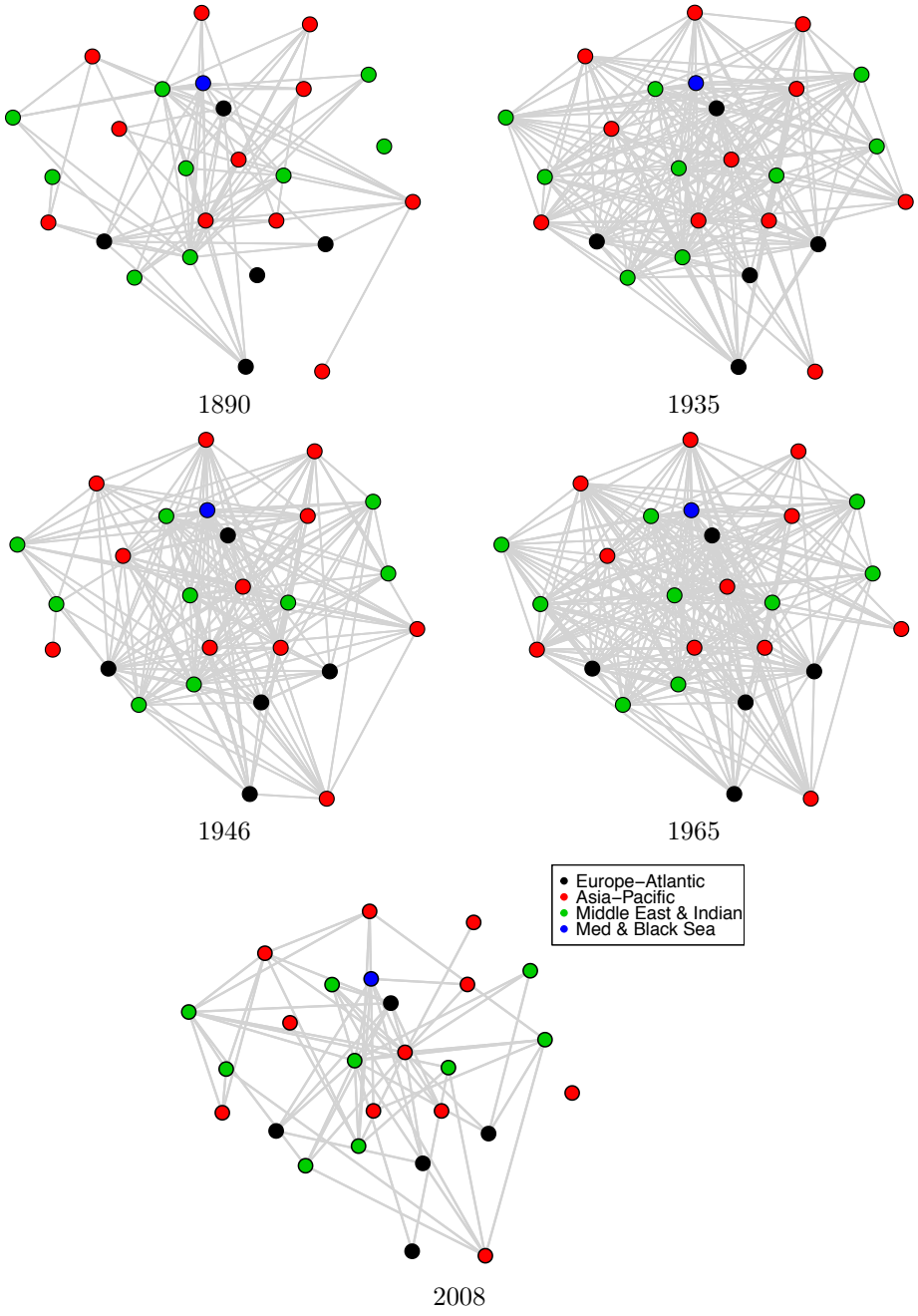


Figure 3.1: Connections between a subset of 26 ports (from October 1890 to October 2008). Data extracted from Lloyd's list. The known subgraphs correspond to geographical regions (continents) indicated using colors.

variables $Z_{iq}^{(t)}$ such that $Z_{iq}^{(t)} = 1$ if at time t, the node i belongs to the class q, and 0 otherwise. Please note that N, C, $\mathcal{P}$, S and Q are all assumed to be constant over time.

3.2.2 *The model at each time* t

As in the original RSM model, the (known) subgraphs are assumed to be built from Q unobserved clusters of vertices, with varying proportions. Thus, each subgraph s has its own mixing proportion vector $\alpha_s^{(t)} = (\alpha_{s1}^{(t)}, \ldots, \alpha_{sQ}^{(t)})$ where $\alpha_{sq}^{(t)}$ is the proportion of cluster k in subgraph s at time t and $\sum_{q=1}^{Q} \alpha_{sq}^{(t)} = 1, \forall s, t$. The network is then assumed to be generated at each time t as follows.

Each vertex i is first associated to a latent cluster q with a probability depending on its subgraph s_i. In practice, the variable $Z_i^{(t)}$ is drawn from a multinomial distribution of parameter $\alpha_{s_i}^{(t)}$:

$$Z_i^{(t)} \sim \mathcal{M}(1, \alpha_{s_i}^{(t)}),$$

and therefore $\sum_{q=1}^{Q} Z_{iq}^{(t)} = 1$. Note that $Z_{iq}^{(t)} = 1$ indicates that vertex i belongs to cluster q at time t, 0 otherwise.

On the other hand, the type of link between nodes i and j is assumed to be sampled from a multinomial distribution depending on the latent vectors $Z_i^{(t)}$ and $Z_j^{(t)}$:

$$X_{ij}^{(t)} | Z_{iq}^{(t)} Z_{jl}^{(t)} = 1 \sim \mathcal{M}(1, \Pi_{ql}),$$

with $\Pi_{ql} \in [0,1]^{C+1}$ and $\sum_{c=0}^{C} \Pi_{ql}^{c} = 1, \forall q, l$.

As in the RSM model, and more generally in SBM like models, all vectors $Z_i^{(t)}$ are sampled independently, and, conditionally on these membership vectors, the edges are assumed to be independent. Thus, contrary to the original RSM model, the edges depend directly on the latent clusters exclusively, and there is no direct dependency on the subgraphs (see Figure 3.3). Each edge between a pair (i, j) of vertices does depend on the subgraphs s_i and s_j, but only through the fact that the edge depends on the latent clusters of the vertices, which themselves depend on the subgraphs. The dependency is indirect while in the original RSM model, the latent clusters along with the subgraphs are all involved in the creation of edges and have different roles. Indeed, the presence or absence of an edge between (i, j) is first drawn from a Bernoulli distribution depending on s_i and s_j. If an edge is present, the edge type is then sampled depending on the latent clusters. The separation of roles between the latent clusters and the subgraphs was originally motivated by assumptions regarding the nature of the networks analyzed. We do not make such assumptions in this paper. The latent clusters explain both the creation of an edge and its type.

Figure 3.2 presents an example of a dRSM network, observed at time t, made of 9 nodes belonging to 2 subgraphs (denoted through the form of nodes) and split into 3 clusters (indicated by the colors).

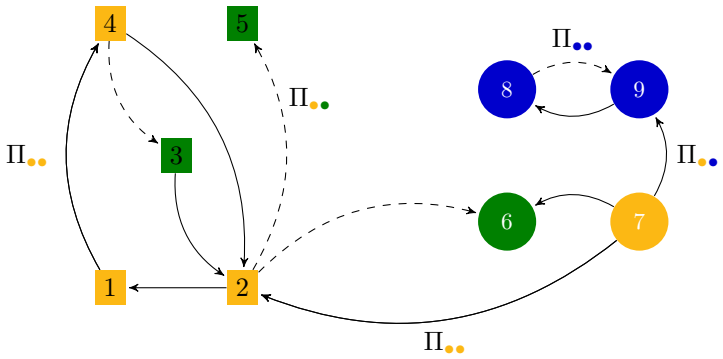


Figure 3.2: A dRSM network observed at time t. The network is made of 9 nodes belonging to $S = 2$ subgraphs (denoted through the form of the nodes) and split into $Q = 3$ clusters (indicated by the colors). According to the dRSM model, the directed edges between the nodes can be of different types ($C = 2$ types are considered here). Given the clusters, the presence of an edge depends on the connection probabilities between clusters (Π).

3.2.3 *Modeling the evolution of random subgraphs*

In order to model the evolution of the cluster proportions within the subgraphs through time, a state space model is considered as in Xing et al. (2010). Thus, the latent variable $\gamma_s^{(t)}$ is introduced and a logistic transformation $f(\cdot)$ is used to link the mixing vector $\alpha_s^{(t)}$ with $\gamma_s^{(t)}$:

$$\alpha_s^{(t)} = f(\gamma_s^{(t)}),$$

such that

$$\alpha_{sq}^{(t)} = f_q(\gamma_s^{(t)}) = \exp(\gamma_{sq}^{(t)} - C(\gamma_s^{(t)})), \forall s, q, t,$$

where $\gamma_{sQ}^{(t)} = 0$ and $C(\gamma_s^{(t)}) = \log(\sum_{q=1}^{Q} \exp(\gamma_{sq}^{(t)}))$. The choice to fix the last component of the vector $\gamma_s^{(t)}$ arbitrarily to 0 is widely used in the literature (see for instance Blei and Lafferty, 2007b; Lafferty and Blei, 2006; Blei and Lafferty, 2007a; Xing et al., 2010) and is due to the bijectivity constraint of this logistic transformation which requires $\gamma_s^{(t)}$ to live in a $(Q-1)$ dimensional space since $\alpha_s^{(t)}$ has $(Q-1)$ degrees of freedom. This induces that $\gamma_{sq}^{(t)} = \log(\alpha_{sq}^{(t)}/\alpha_{sQ}^{(t)}), \forall s, q, t$. In addition, the $(Q-1)$ first components of the vector $\gamma_s^{(t)}$ are assumed to be distributed according to a Gaussian distribution with mean $B\nu^{(t)}$ and covariance matrix Σ:

$$\gamma_{s\backslash Q}^{(t)} \sim \mathcal{N}(B\nu^{(t)}, \Sigma), \tag{3.1}$$

where $\gamma_{s\backslash Q}^{(t)}$ is the vector $\gamma_s^{(t)}$ without his last component. Both Σ and B are matrices of size $(Q-1) \times (Q-1)$ while $\nu^{(t)}$ is a $(Q-1)$ dimensional vector. Let

us notice that even though the $\gamma_s^{(t)}$ have the same mean in the state-space, they are actually independent and thus play different roles.

The rest of the model now involves a classic state space model for linear dynamic systems. It is defined as follows:

$$\begin{cases} \nu^{(t)} = A\nu^{(t-1)} + \omega \\ \nu^{(1)} = \mu_0 + u. \end{cases}$$

The noise terms ω and u are supposed to be Gaussian and independent:

$$\begin{cases} \omega \sim \mathcal{N}(0, \Phi) \\ u \sim \mathcal{N}(0, V_0). \end{cases}$$

A and B matrices of transition. Notice that the state space model for linear dynamic systems may suffer from model identifiability issues and constraints have to be introduced (see for instance Harvey, 1989). In the following, we derive the inference procedure in a general context since different constraints can be considered. In practice, in all the experiments that we carried out, we fixed A, B, and V_0 to be equal to the identity matrix I_{Q-1} and all components of μ_0 to zero.

The model described here has three sets of latent variables ($\nu = (\nu^{(t)})_t, \gamma = (\gamma_s^{(t)})_{st}, Z = (Z_{iq}^{(t)})_{iqt}$) and is parameterized by $\theta = (\mu_0, A, B, \Phi, V_0, \Sigma, \Pi)$. Note that all parameters in θ depend neither on time nor subgraphs. This model is called the *dynamic random subgraph model* (dRSM) in the rest of the document. Figure 3.3 gives the graphical model for dRSM and Table 3.1 summarizes the notations used in the model.

At this point, it is possible to see some links and differences between dRSM and dM3SBM (Ho et al., 2011), which is the closest model in the litterature. On the one hand, dRSM and dM3SBM share a common way to model the latent clusters and the temporal dynamic through a state space model. On the other hand, dRSM is able to handle categorical edges, which is a useful feature when working on real-world networks, whereas dM3SBM cannot. In addition, dRSM requires the knowledge of the subgraphs whereas dM3SBM proposes to estimate them. Furthermore, dM3SBM allows the nodes to belong to different clusters. However, allowing to estimate the subgraphs and multi-group belongings may conduce dM3SBM to be a too flexible model and thus to fail in recovering the network structure. Indeed, providing the subgraphs to dRSM allows it to avoid looking for obvious structures such that it can focus on the search of hidden patterns. The comparisons presented in Section 3.4 seem to confirm this thesis.

3.2.4 Joint distribution of dRSM

The dRSM model proposed above is defined by the joint distribution:

$$p(X, Z, \gamma, \nu|\theta) = p(X|Z, \Pi)p(Z|\gamma)p(\gamma_{\setminus Q}|B, \nu, \Sigma)p(\nu|\mu_0, A, \Phi, V_0), \qquad (3.2)$$

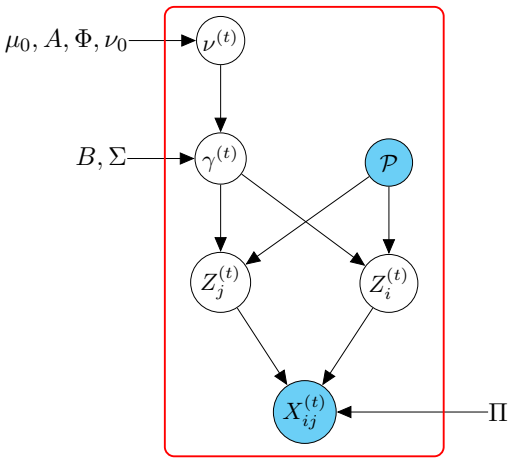


Figure 3.3: *Graphical representation of the dRSM model.*

where $\gamma_{\backslash Q} = (\gamma_{s\backslash Q}^{(t)})_{st}$. Moreover

$$p(X|Z,\Pi) = \prod_{t=1}^{T}\prod_{q,l}^{Q}\prod_{c=0}^{C}(\Pi_{ql}^{c})^{\sum_{i\neq j}^{N}\delta(X_{ij}^{(t)}=c)Z_{iq}^{(t)}Z_{jl}^{(t)}},$$

and

$$\begin{aligned} p(Z|\gamma) &= \prod_{t=1}^{T}\prod_{i=1}^{N}\prod_{q=1}^{Q} f_k(\gamma_{s_i}^{(t)})^{Z_{iq}^{(t)}} \\ &= \prod_{t=1}^{T}\prod_{q=1}^{Q}\prod_{s=1}^{S} f_k(\gamma_{s}^{(t)})^{\sum_{i=1}^{N} y_{is} Z_{iq}^{(t)}}. \end{aligned} \tag{3.3}$$

Note that

$$p(\gamma_{\backslash Q}|B,\nu,\Sigma) = \prod_{t=1}^{T}\prod_{s=1}^{S}\mathcal{N}(\gamma_{s\backslash Q}^{(t)}; B\nu^{(t)}, \Sigma),$$

where $\mathcal{N}(\gamma_{s\backslash Q}^{(t)}; B\nu^{(t)}, \Sigma)$ denotes the multivariate Gaussian distribution, with mean vector $B\nu^{(t)}$ and covariance matrix Σ, evaluated at $\gamma_{s\backslash Q}^{(t)}$. Finally

$$p(\nu|\mu_0, A, \Phi, V_0) = p(\nu^{(1)}|\mu_0, V_0)\prod_{t=2}^{T}\log p(\nu^{(t)}|\nu^{(t-1)}, A, \Phi).$$

3.3 Estimation

This section focuses on the inference of the model proposed above. A variational EM algorithm is considered and a model selection criterion is derived.

Notations	Description
X	Adjacency matrix $X_{ij}^{(t)} \in \{0, \dots, C\}$ at each t
Z	Binary matrix. $Z_{iq}^{(t)} = 1$ indicates that i belongs to cluster k at t
N	Number of vertices in the network
Q	Number of latent clusters
S	Number of subgraphs
C	Number of edge types
Π	Π_{ql}^{c} is the probability of having an edge of type c between vertices of clusters k and l
α	$\alpha_{sq}^{(t)} = f_q(\gamma_s^{(t)})$ is the proportion of cluster q in the subgraph s at t

Table 3.1: *Summary of the notations used in the chapter.*

3.3.1 A variational framework

We aim at maximizing the log-likelihood $\log p(X|\theta)$ associated with the model. To achieve this maximization, a common approach consists in using an expectation maximization (EM) algorithm (Dempster et al., 1977; Krishnan and McLachlan, 1997). However, such an algorithm cannot be derived here since $p(Z, \gamma, \nu|X, \theta)$ is intractable. Therefore, we propose to use a variational EM-type algorithm (VEM) (Hathaway, 1986) which locally optimizes the model parameters with respect to a lower bound of the log-likelihood. Thus, given a distribution q for the three sets of latent variables (Z, γ, ν), the log-likelihood can be written:

$$\log p(X|\theta) = \mathcal{L}(q, \theta) + KL(q(.) \parallel p(.|X, \theta)), \tag{3.4}$$

where $\mathcal{L}$ is defined as follows:

$$\mathcal{L}(q, \theta) = \sum_{Z} \int_{\gamma} \int_{\nu} q(Z, \gamma, \nu) \log \frac{p(X, Z, \gamma, \nu|\theta)}{q(Z, \gamma, \nu)} d\gamma \, d\nu, \tag{3.5}$$

and KL denotes the Kullback-Leibler divergence between the true and approximate posterior distributions:

$$KL(q(.) \parallel p(.|X, \theta)) = -\sum_{Z} \int_{\gamma} \int_{\nu} q(Z, \gamma, \nu) \log \frac{p(Z, \gamma, \nu|X, \theta)}{q(Z, \gamma, \nu)} d\gamma \, d\nu. \tag{3.6}$$

Looking for the best approximation of the posterior distribution $p(Z, \gamma, \nu|X, \theta)$ in the sense of the KL divergence becomes equivalent to searching for a distribution $q(\cdot)$ that maximizes the lower bound $\mathcal{L}$ of the integrated log-likelihood. Unfortunately, because the joint distribution (3.2) in the lower bound involves the quantity $p(Z|\gamma)$ which depends on the normalizing constant $C(\gamma_s^{(t)})$, $\mathcal{L}$ has no analytical form and cannot be optimized with respect to $q(\cdot)$. Indeed, $C(\gamma_s^{(t)}) = \log(\sum_{l=1}^{Q} \exp(\gamma_{sl}^{(t)}))$ is based on a non linear transformation of the vector $\gamma_s^{(t)}$ which makes some expectations of the standard VEM algorithm impossible to derive.

Following the work of Lafferty and Blei (2006) on correlated topic models, we propose a new bound of $\mathcal{L}(q,\theta)$ based on a variational lower bound of $p(Z|\gamma)$, as in Jordan et al. (1999).

Proposition 3.3.1 *Given any set ξ of variational parameters $\xi_s^{(t)} \in \mathbb{R}^{*+}$, a lower bound of the first lower bound $\mathcal{L}(q,\theta)$ is given by:*

$$\log p(X|\theta) \geqslant \mathcal{L}(q,\theta) \geqslant \tilde{\mathcal{L}}(q,\theta,\xi),$$

where

$$\begin{aligned}&\tilde{\mathcal{L}}(q,\theta,\xi)\\ &= \sum_Z \int_\gamma \int_\nu q(Z,\gamma,\nu) \log \frac{p(X|Z,\Pi)h(Z,\gamma,\xi)p(\gamma_{\setminus Q}|B,\nu,\Sigma)p(\nu|\mu_0,A,\Phi,V_0)}{q(Z,\gamma,\nu)} d\gamma d\nu\end{aligned} \tag{3.7}$$

with

$$\log h(Z,\gamma,\xi) = \sum_{t=1}^{T}\sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{s=1}^{S} y_{is} Z_{iq}^{(t)} \Big(\gamma_{sq}^{(t)} - \big(\xi_s^{-1(t)} \sum_{l=1}^{Q} \exp(\gamma_{sl}^{(t)}) - 1 + \log(\xi_s^{(t)})\big)\Big).$$

***Proof of Prop. 3.3.1*:** We rely on a bound introduced in Jordan et al. (1999). Such a general bound can easily been derived by noticing that $C(\cdot)$ is a concave function of $\sum_{l=1}^{K} \exp(\gamma_{s_i l}^{(t)})$ and therefore a first order Taylor expansion of the normalizing constant, at any $\xi_s^{(t)} \in \mathbb{R}^{*+}$, will lead to the inequality:

$$\log(\sum_{l=1}^{K} \exp(\gamma_{sl}^{(t)})) \leq \xi_s^{-1(t)} (\sum_{l=1}^{K} \exp(\gamma_{sl}^{(t)})) - 1 + \log(\xi_s^{(t)}). \tag{3.8}$$

The bounds (3.8) on the $C(\gamma_s^{(t)})$ terms induce a lower bound on the quantity $\log p(Z|\gamma)$:

$$\begin{aligned}\log p(Z|\gamma) &= \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=1}^{N} Z_{ik}^{(t)} \log(f_k(\gamma_{s_i}^{(t)}))\\ &= \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s=1}^{S} y_{is} Z_{ik}^{(t)} \Big(\gamma_{sk}^{(t)} - \log(\sum_{l=1}^{K} \exp(\gamma_{sl}^{(t)}))\Big)\\ &\geq \log h(Z,\gamma,\xi),\end{aligned}$$

where ξ denotes the set of all variational parameters $(\xi_s^{(t)})_{st}$ and the function $h(\cdot,\cdot,\cdot)$ is such that:

$$\log h(Z,\gamma,\xi) = \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s=1}^{S} y_{is} Z_{ik}^{(t)} \Big(\gamma_{sk}^{(t)} - \big(\xi_s^{-1(t)} \sum_{l=1}^{K} \exp(\gamma_{sl}^{(t)}) - 1 + \log(\xi_s^{(t)})\big)\Big).$$

Replacing $\log p(Z|\gamma)$ by $\log h(Z, \gamma, \xi)$ in $\mathcal{L}(q, \theta)$, leads to a new lower bound $\tilde{\mathcal{L}}(q, \theta, \xi)$ for $\log p(X|\theta)$ which satisfies:

$$\log p(X|\theta) \geqslant \mathcal{L}(q, \theta) \geqslant \tilde{\mathcal{L}}(q, \theta, \xi). \quad \blacksquare$$

Note that the variational parameters $\xi_s^{(t)}$ can be optimized to obtain tight bounds (see the end of Section 3.3.2). Moreover, we emphasize that a variational parameter $\xi_s^{(t)}$ is considered for each subgraph s and each time t for more flexibility and to improve the inference procedure. We point out that the quality of the variational approximation we propose cannot be tested analytically since $\tilde{\mathcal{L}}(q, \theta, \xi)$ and the Kullback-Leibler divergence in (3.6) are not tractable. Nevertheless, we rely on them for inference purposes. Note that similar approximation schemes have been used for instance by Svensén and Bishop (2004) and Latouche et al. (2014), in the context of model selection.

In order to maximize $\tilde{\mathcal{L}}(q, \theta, \xi)$, we further assume that $q(Z, \gamma, \nu)$ can be factorized:

$$q(Z, \gamma, \nu) = q(Z)q(\gamma)q(\nu) = \Big(\prod_{t=1}^{T}\prod_{i=1}^{N} q(Z_i^{(t)})\Big)q(\gamma)q(\nu).$$

Finally $q(\gamma)$ is chosen within the family of Gaussian distributions of the form:

$$q(\gamma) = \prod_{t=1}^{T}\prod_{s=1}^{S}\prod_{q=1}^{Q} \mathcal{N}(\gamma_{sq}^{(t)}; \hat{\gamma}_{sq}^{(t)}, \hat{\sigma}_{sq}^{2^{(t)}}),$$

to derive analytical expectations in the E step, as in Lafferty and Blei (2006). Since the last component of each vector $\gamma_s^{(t)}$ has to remain equal to zero, to preserve the bijectivity constraints of the transformation $f(\cdot)$, the terms $\hat{\gamma}_{sQ}^{(t)}$ and $\hat{\sigma}_{sQ}^{2^{(t)}}$ are all set to zero to ensure a Dirac mass at zero. All other mean and variance terms $(\hat{\gamma}_{sq}^{(t)}, \hat{\sigma}_{sq}^{2^{(t)}}), \forall s, k \neq Q, t$, are parameters to be estimated.

3.3.2 A VEM algorithm for the dRSM model

In this section, we first assume that the variational terms ξ, which were introduced for approximation purposes, are given. This allows the use of a VEM algorithm (Jordan et al., 1999) to maximize the lower bound $\tilde{\mathcal{L}}(q, \theta, \xi)$ with respect to $q(Z, \gamma, \nu)$ and the model parameters θ (a sketch is given by the algorithm 4). Such an optimization procedure is iterative and involves a series of successive updates. In the E step, the model parameters are fixed and the lower bound is optimized with respect to $q(Z, \gamma, \nu)$. Conversely, during the M step, the variational distribution is held fixed while $\tilde{\mathcal{L}}(q, \theta, \xi)$ is maximized with respect to θ. In standard VEM algorithms, a unique set of latent variables is usually considered. In our case, there are three sets (Z, γ, ν) of latent variables and therefore the E step itself involves iterative updates (as in Latouche et al., 2014, for instance). All distributions in $q(Z, \gamma, \nu)$ are held fixed, except one, which is optimized. This procedure is repeated for all distributions in turn.

In the following, we give in details the update formulate for the E and M steps.

Proposition 3.3.2 *The VEM update step for each distribution $q(Z_i^{(t)})$ is given by:*

$$q(Z_i^{(t)}) \sim \mathcal{M}\left(Z_i^{(t)}; 1, \tau_i^{(t)} = (\tau_{i1}^{(t)}, \ldots, \tau_{iQ}^{(t)})\right) \ \forall i, t,$$

where

$$\tau_{iq}^{(t)} \propto \exp\left(\sum_{l=1}^{Q}\sum_{c=0}^{C}\sum_{i\neq j}^{N}\delta(X_{ij}^{(t)}=c)\tau_{jl}^{(t)}\Big[\log(\Pi_{ql}^{c})+\log(\Pi_{lq}^{c})\Big] + \sum_{s=1}^{S} y_{is}\Big(\hat{\gamma}_{sq}^{(t)} - \Big(\xi_s^{-1(t)}\sum_{l=1}^{Q}\exp(\hat{\gamma}_{sl}^{(t)}+\frac{\hat{\sigma}_{sl}^{2^{(t)}}}{2}) - 1 + \log(\xi_s^{(t)})\Big)\Big)\right).$$

Proof of Prop. 3.3.2:

$$\begin{aligned}
\log q(Z_i) &= E_{\gamma,\nu,Z^{\setminus i}}[\log p(X|Z,\Pi) + \log h(Z,\gamma,\xi)] + \text{const} \\
&= \sum_{t=1}^{T}\sum_{k=1}^{K} Z_{ik}^{(t)}\left(\sum_{l=1}^{K}\sum_{c=0}^{C}\sum_{i\neq j}^{N}\delta(X_{ij}^{(t)}=c)\tau_{jl}^{(t)}\Big[\log(\Pi_{kl}^{c})+\log(\Pi_{lk}^{c})\Big]\right) \\
&+ \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{s=1}^{S} y_{is}E_{\gamma}\Big[Z_{ik}^{(t)}\log h(Z^{(t)},\gamma^{(t)},\xi^{(t)})\Big] + \text{const.} \\
&= \sum_{t=1}^{T}\sum_{k=1}^{K} Z_{ik}^{(t)}\left(\sum_{l=1}^{K}\sum_{c=0}^{C}\sum_{i\neq j}^{N}\delta(X_{ij}^{(t)}=c)\tau_{jl}^{(t)}\Big[\log(\Pi_{kl}^{c})+\log(\Pi_{lk}^{c})\Big]\right) \\
&+ \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{s=1}^{S} y_{is}E_{\gamma}\Big[\gamma_{sk}^{(t)} - (\xi_s^{-1(t)}\sum_{l=1}^{K}\exp(\gamma_{sk}^{(t)}) - 1 + \log(\xi_s^{(t)}))\Big] + \text{const.} \\
&= \sum_{t=1}^{T}\sum_{k=1}^{K} Z_{ik}^{(t)}\left(\sum_{l=1}^{K}\sum_{c=0}^{C}\sum_{i\neq j}^{N}\delta(X_{ij}^{(t)}=c)\tau_{jl}^{(t)}\Big[\log(\Pi_{kl}^{c})+\log(\Pi_{lk}^{c})\Big]\right) \\
&+ \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{s=1}^{S} Z_{ik}^{(t)}y_{is}\left(\hat{\gamma}_{sk}^{(t)} - \Big[\xi_s^{-1(t)}\sum_{l=1}^{K}E(\exp(\gamma_{sl}^{(t)})) - 1 + \log(\xi_s^{(t)})\Big]\right) + \text{const.} \\
&= \sum_{t=1}^{T}\sum_{k=1}^{K} Z_{ik}^{(t)}\left(\sum_{l=1}^{K}\sum_{c=0}^{C}\sum_{i\neq j}^{N}\delta(X_{ij}^{(t)}=c)\tau_{jl}^{(t)}\Big[\log(\Pi_{kl}^{c})+\log(\Pi_{lk}^{c})\Big]\right. \\
&\left. + \sum_{s=1}^{S} y_{is}\Big(\hat{\gamma}_{sk}^{(t)} - \Big(\xi_s^{-1(t)}\sum_{l=1}^{K}\exp(\hat{\gamma}_{sl}^{(t)}+\frac{\hat{\sigma}_{sl}^{2^{(t)}}}{2}) - 1 + \log(\xi_s^{(t)})\Big)\Big)\right) + \text{const},
\end{aligned}$$

where all terms that do not depend on Z_i have been put into the constant terms

const. Moreover since $\gamma_{sk}^{(t)} \sim \mathcal{N}(\hat{\gamma}_{sk}^{(t)}, \hat{\sigma}_{sk}^{2^{(t)}})$ we have used:

$$\mathbb{E}[\exp(\gamma_{sk}^{(t)})] = \exp(\hat{\gamma}_{sk}^{(t)} + \frac{\hat{\sigma}_{sk}^{2^{(t)}}}{2}).$$

We then recognize the functional form of a multinomial distribution:

$$q(Z_i^{(t)}) \sim \mathcal{M}(Z_i^{(t)}; 1, \tau_i^{(t)}), \ \forall i, t. \quad \blacksquare$$

Note that $\tau_{iq}^{(t)}$ is the approximate posterior probability that node i belongs to cluster q at time t.

Proposition 3.3.3 *The VEM update step for the distribution $q(\nu)$ is given by:*

$$q(\nu) \propto p(\nu^{(1)}|\mu_0, V_0)\Big[\prod_{t=2}^{T} p(\nu^{(t)}|\nu^{(t-1)}, A, \Phi)\Big]\Big[\prod_{t=1}^{T} \mathcal{N}(\frac{\sum_{s=1}^{S}\hat{\gamma}_s^{(t)}}{S}; B\nu^{(t)}, \frac{\Sigma}{S})\Big].$$

Proof of Prop. 3.3.3:

$$\begin{aligned}
\log q(\nu) &= E_{Z,\gamma}\Big(\log p(\gamma|\nu, \Sigma, B) + \log p(\nu|\mu_0, V_0, A, \Phi)\Big) + \text{const} \\
&= \sum_{t=1}^{T}\sum_{s=1}^{S}\Big(E_\gamma\big(\log \mathcal{N}(\gamma_s^{(t)}; B\nu^{(t)}, \Sigma)\big)\Big) + \log p(\nu^{(1)}|\mu_0, V_0) \\
&+ \sum_{t=2}^{T}\log p(\nu^{(t)}|\nu^{(t-1)}, A, \Phi) + \text{const} \\
&= \sum_{t=1}^{T}\sum_{s=1}^{S}\Big(E_\gamma\big(-\frac{1}{2}(\gamma_s^{(t)})^\intercal\Sigma^{-1}(\gamma_s^{(t)}) + (\gamma_s^{(t)})^\intercal\Sigma^{-1}B\nu^{(t)} - \frac{1}{2}(\nu^{(t)})^\intercal B^\intercal\Sigma^{-1}B\nu^{(t)}\big)\Big) \\
&+ \log p(\nu^{(1)}|\mu_0, V_0) + \sum_{t=2}^{T}\log p(\nu^{(t)}|\nu^{(t-1)}, A, \Phi) + \text{const.} \\
&= \sum_{t=1}^{T}\Big(\sum_{s=1}^{S}\Big(\hat{\gamma}_s^{(t)}\Sigma^{-1}B\nu^{(t)}\Big) - \frac{1}{2}(\nu^{(t)})^\intercal B^\intercal(S\Sigma^{-1})B\nu^{(t)}\Big) \\
&+ \log p(\nu^{(1)}|\mu_0, V_0) + \sum_{t=2}^{T}\log p(\nu^{(t)}|\nu^{(t-1)}, A, \Phi) + \text{const},
\end{aligned}$$

where all terms that do not depend on ν have been put into the constant terms const. We recognize the functional form of the posterior distribution of a linear dynamic system:

$$\begin{aligned}
\log q(\nu) &= \sum_{t=1}^{T}\Big(\log \mathcal{N}(\frac{\sum_{s=1}^{S}\hat{\gamma}_s^{(t)}}{S}; B\nu^{(t)}, \frac{\Sigma}{S})\Big) \\
&+ \log p(\nu^{(1)}|\mu_0, V_0) + \sum_{t=2}^{T}\log p(\nu^{(t)}|\nu^{(t-1)}, A, \Phi) + \text{const.} \quad \blacksquare
\end{aligned}$$

At this step, we recall that the terms $\hat{\gamma}_s^{(t)}$ are fixed and so is the variable $x^{(t)} = \sum_{s=1}^{S} \hat{\gamma}_s^{(t)}/S$. Therefore, it is remarkable to note that the functional form of $q(\nu)$ corresponds exactly to the form of the posterior distribution associated with a state space model where ν is the set of all latent state variables and $x = (x^{(t)})_t$ the set of observed outputs. Thus, each $x^{(t)}$ can be written as $x^{(t)} = B\nu^{(t)} + \tilde{v}$ where $\tilde{v} \sim \mathcal{N}(0, \Sigma/S)$ while the variables in ν are defined as previously:

$$\begin{cases} \nu^{(t)} = A\nu^{(t-1)} + \omega \\ \nu^{(1)} = \mu_0 + u, \end{cases}$$

with

$$\begin{cases} \omega \sim \mathcal{N}(0, \Phi) \\ u \sim \mathcal{N}(0, V_0). \end{cases}$$

Contrary to the original state space model introduced in Section 3.2, where both γ and ν were sets of unobserved variables, we obtain here a standard linear dynamic system from which the corresponding parameters, *i.e.* $\theta' = (\mu_0, A, B, \Phi, V_0, \Sigma/S)$ can be estimated using Kalman filter and Rauch-Tung-Striebel (RTS) smoother (Rauch et al., 1965b) (details can also be found in Minka, 1998). The expectations $\hat{\nu}^{(t)}$ and covariance matrices $\hat{V}^{(t)}$ of the random variables $\nu^{(t)}$, given all the observed data x, are determined relying on backward forward recursions.

Proposition 3.3.4 *After the E step of the VEM algorithm, the lower bound* $\tilde{\mathcal{L}}(q, \theta, \xi)$ *simplifies into:*

$$\begin{aligned}
\tilde{\mathcal{L}}(q,\theta,\xi) \quad = \quad & \sum_{t=1}^{T}\sum_{q,l}\sum_{c=0}^{C}\sum_{i\neq j}^{N} \delta(X_{ij}^{(t)} = c)\tau_{iq}^{(t)}\tau_{jl}^{(t)} \log(\Pi_{ql}^{c}) \\
+ \quad & \sum_{t=1}^{T}\sum_{s=1}^{S}\Big(r_s^{(t)}\hat{\gamma}_{sq}^{(t)} - N_s\xi_s^{-1(t)}\sum_{l=1}^{Q}\exp(\hat{\gamma}_{sl}^{(t)} + \frac{\hat{\sigma}_{sl}^{2^{(t)}}}{2}) + N_s - N_s\log(\xi_s^{(t)})\Big) \\
+ \quad & \sum_{t=1}^{T}\sum_{s=1}^{S}\Big(\log\mathcal{N}(\hat{\gamma}_s^{(t)}, B\hat{\nu}_s^{(t)}, \Sigma) - \frac{1}{2}tr(\Sigma^{-1}B^{\intercal}\hat{V}^{(t)}B) - \frac{1}{2}tr(\Sigma^{-1}\hat{\sigma}_s^{(t)^2})\Big) \\
- \quad & \sum_{t=1}^{T}\sum_{s=1}^{S}\sum_{q=1}^{Q-1} -\log\Big((2\pi)^{\frac{1}{2}}\hat{\sigma}_{sq}^{(t)}\Big) + \frac{TQS}{2} \\
- \quad & \sum_{t=1}^{T}\Big(\log\mathcal{N}(x^{(t)}; B\hat{\nu}^{(t)}, \frac{\Sigma}{S}) + \frac{1}{2}tr(\Sigma^{-1}SB^{\intercal}\hat{V}^{(t)}B)\Big) \\
- \quad & \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{q=1}^{Q}\tau_{iq}^{(t)}\log(\tau_{iq}^{(t)}) \\
+ \quad & \log p(x|\theta')
\end{aligned}$$

where $r_s^{(t)} = \sum_{i=1}^{N}\tau_{iq}^{(t)}y_{is}$, N_s *is a number of nodes in the subgraph* s, *and* $\log p(x|\theta')$ *is the log likelihood of the linear dynamic system associated with the variational distribution* $q(\nu)$ *(see Appendix B).*

The maximization of this bound allows to obtain the updating formula for the tensor matrix Π:

$$\hat{\Pi}^c_{ql} = \frac{\sum_{t=1}^{T}\sum_{i\neq j}^{N}\delta(X_{ij}^{(t)}=c)\tau_{iq}^{(t)}\tau_{jl}^{(t)}}{\sum_{t=1}^{T}\sum_{c=0}^{C}\sum_{i\neq j}^{N}\delta(X_{ij}^{(t)}=c)\tau_{iq}^{(t)}\tau_{jl}^{(t)}}, \forall q,l,c.$$

For the parameters $\hat{\gamma}_{sq}^{(t)}$ and $\hat{\sigma}_{sq}^{2^{(t)}}$, we do not obtain analytical expressions, and therefore we rely on a quasi-Newton algorithm for the optimization task.

3.3.3 Optimization of ξ

So far, we have seen that a VEM algorithm could be implemented from approximations depending on the variational parameters $\xi_s^{(t)}$. However, we have not addressed yet how these parameters could be estimated from the data. We follow the work of Svensén and Bishop (2004) on Bayesian hierarchical mixture of experts. Thus, the lower bound $\tilde{\mathcal{L}}(q,\theta,\xi)$ is optimized with respect to the variational terms $\xi_s^{(t)}$ to obtain the tightest bound $\tilde{\mathcal{L}}(q,\theta,\xi)$ of $\mathcal{L}(q,\theta)$. This leads to new estimates $\hat{\xi}_s^{(t)}$ of $\xi_s^{(t)}$:

$$\hat{\xi}_s^{(t)} = \sum_{l=1}^{Q}\exp(\hat{\gamma}_{sl}^{(t)} + \hat{\sigma}_{sl}^{2^{(t)}}), \forall s,t.$$

This procedure gives rise to a three step optimization scheme. Given all $\xi = (\xi_s^{(t)})_{st}$, the VEM algorithm described previously is used to maximize the lower bound with respect to $q(Z,\gamma,\nu)$ and θ. These terms are then held fixed and a new estimate of ξ is computed. The three steps are repeated until convergence of the lower bound.

Algorithme 4: VEM algorithm for the dRSM model in which we have Q latent groups.

Initialization of $\theta^0 = (\mu_0, A, \Phi, V_0, \Pi, B, \Sigma)$
Initialization of the matrix τ at each instant t
Found $\hat{\nu}, \hat{V}, (\hat{\gamma}_{sq}^{(t)}, \hat{\sigma}_{sq}^{2^{(t)}})_{sq}^{t}, \hat{\theta}, \hat{\xi}$
Calculate $\tilde{\mathcal{L}}(q,\hat{\theta},\hat{\xi})$
While $\mid \tilde{\mathcal{L}}^{new} - \tilde{\mathcal{L}}^{old} \mid > \varepsilon$
    Update of $\tau, \hat{\nu}, \hat{V}, (\hat{\gamma}_{sq}^{(t)}, \hat{\sigma}_{sq}^{2^{(t)}})_{sqt}$ (E-step)
    Update of $\hat{\theta}$ (M-step)
    Update of $\hat{\xi}$
    Calculate of $\mathcal{L}(q,\hat{\theta},\hat{\xi})$
end of loop

3.3.4 Model selection: choice of the number Q of latent groups

Using the VEM algorithm proposed in the previous paragraphs, the estimation of the model parameters and of the group memberships is fully automatic for a given value of Q. Since we consider here a model-based approach, two dRSM models with different values of Q can be considered as two different models. The problem of choosing Q can therefore be viewed as a model selection problem. It can be tackled in a model-based context using model selection criteria, such as the Akaike information criterion (AIC) (Akaike, 1974) or the Bayesian information criterion (BIC) (Schwarz, 1978). Due to its popularity and its asymptotic properties (Leroux, 1992), we use BIC in the numerical experiments presented in the following sections. BIC relies on an asymptotic approximation of the marginal log-likelihood, also called integrated log-likelihood, and is defined in the specific context of the dRSM model $\mathcal{M}$ by:

$$BIC(\mathcal{M}) = \log p(X|\hat{\theta}) - \frac{\eta(\mathcal{M})}{2} \log(TN(N-1)),$$

where $\eta(\mathcal{M})$ is the number of free model parameters depending on Q, for the identifiability constraints considered. Unfortunately, the log-likelihood $\log p(X|\hat{\theta}) = \log\left(\sum_Z p(X, Z|\hat{\theta}\right)$ is not tractable here because it involves marginalizing over all latent vectors $Z_i^{(t)}$ in Z. Therefore, we propose to replace the log-likelihood with its variational approximation $\tilde{\mathcal{L}}(q, \theta, \xi)$. Thus, the VEM algorithm is run for various values of Q. For each Q, the algorithm iterates until convergence of the lower bound. $\hat{Q}$ is then chosen such that the (approximate) BIC criterion is maximized.

3.4 Numerical experiments and comparisons

This section aims at proving on synthetic data the validity of the inference algorithm presented in section 3. An introductory example is first considered to highlight the main features of the proposed approach. Secondly, we study the influence of the size of the network on the quality of results. Model selection is then, considered to validate the criterion choice. Extensive comparisons with state-of-the-art methods conclude this section.

3.4.1 Experimental setup

In order to validate our approach, we use in this section artificial data generated according to a common experimental setup. To simplify the characterization and facilitate the reproducibility of the experiments, we designed five different scenarios. The generation setup for each scenario is summarized in Table 3.2. Data from scenario 0 are drawn using SBM at each time t and without an explicit temporal dependence. The data sets for all other scenarios (scenarios 1 to 4)

Parameters	Scenario 0	Scenario 1	Scenario 2	Scenario 3	Scenario 4
N	300				
Q	4				
T	10 (indep.)	10 (ssm)			
S	1	1	1	2	2
C	1	1	1	1	2
$(\Pi^0_{ll})_{l=1,..,Q}$	(0.1,0.4,0.5,0.6)				
$\Pi^0_{ql,q\neq l}$	0.99		0.8	0.99	
$\Pi^{c\neq 0}_{ql}$	$(1-\Pi^0_{ql})/C$				

Table 3.2: Parameter values for the five types of graphs used in the experiments. In scenario 0, the networks are drawn without an explicit temporal dependance whereas, in the other scenarios, the temporal dependance is generated through a state space model (ssm).

are drawn according to the dRSM model. Therefore, the temporal dependence is generated through a state space model. All generated networks are made of $N = 300$ nodes, distributed into $Q = 4$ latent groups and have $T = 10$ time points. Depending on the scenario, the networks have $S = 1$ or 2 subgraphs, with binary ($C = 1$) or categorical ($C = 2$) edges. When $S > 1$, the nodes are randomly assigned uniformly to the subgraphs. Notice that scenario 2 has a parameter $\Pi^0_{ql,q\neq l}$ equal to 0.8 which leads to less heterogeneous latent groups.

The model parameters used for the simulation are as follows. For the simulation of γ, it is assumed that the matrices A, B and V_0 are set to I_{Q-1}, and that $\Sigma = 0.1 \times Q \times I_{Q-1}$ and $\Phi = 0.01 \times I_{Q-1}$. Finally, the tensor matrix Π, which defines the connection probabilities between clusters for the C different types, is set up such that, within the clusters, the probability $1 - \Pi^0_{ll}$ of having an edge of any type is larger than the corresponding connection probabilities between clusters $1 - \Pi^0_{ql,q\neq l}$ (see Table 3.2). Notice that such a choice of parameters induces networks made of communities. Then, in case of a connection between two nodes, the edge type is sampled uniformly, *i.e.* $\Pi^{c\neq 0}_{ql} = (1 - \Pi^0_{ql})/C, \ \forall q, l$.

3.4.2 An introductory example

We first focus on an introductory example to illustrate the global behavior of the proposed methodology. To this end, we simulated a single network according to scenario 2 for facilitating the understanding of the results. We remind that in this setup the number Q of latent groups is fixed to 4 and that $C = 1$. Therefore, the network is binary and Π^1_{ql} indicates the occurrence probability of an edge. We ran the VEM algorithm on it for a number Q of groups ranging from 3 to 6. We selected afterward the most appropriate number of groups using the BIC criterion.

Figure 3.4 shows the BIC values associated to the results provided by our VEM algorithm for the different values of Q. One can observe that the criterion picks at $Q = 4$, which is the actual simulated value for Q. Figure 3.5 presents

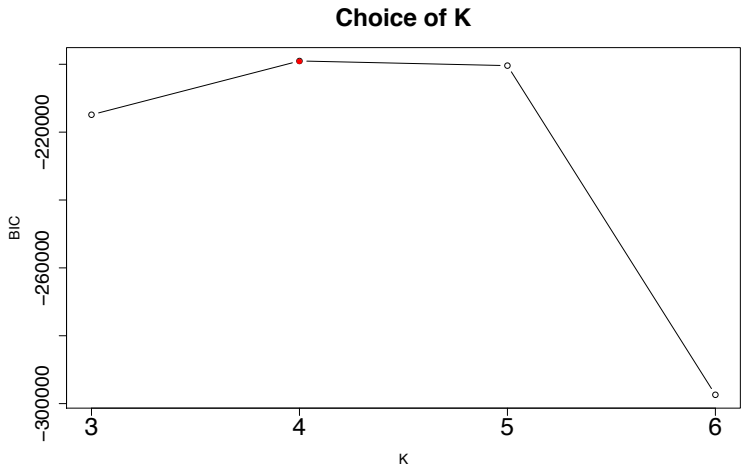


Figure 3.4: Choice of Q by model selection with BIC for a simulated network. The actual value for Q is 4.

the evolution of the bound $\tilde{\mathcal{L}}$ for this specific value of Q along the 10 iterations of the VEM algorithm. A clear plateau of the bound is visible on the figure, which indicates the convergence of the algorithm.

To quickly assess the estimation quality, Table 3.3 allows to compare the actual (left panel) and estimated (right panel) values of the terms Π^1_{ql} in the tensor matrix Π, which define the connection probabilities between the latent clusters. On this single example, the estimated values Π^1_{ql} turn out to be extremely close to the true ones. As well as, we aim the compare the values of the matrix Π in the case where the edges are categorical, to this end we used scenario 4 with $Q = 3$. Table 3.4 allows to compare the actual (left panel) and estimated (right panel) values of the terms Π^c_{ql} where c=(0,1,2), similarly the estimated values in all values of c appears very close to true values of Π. Finally, Figure 3.6 compares the actual (dashed red lines) and estimated (solid black lines) values of the group proportions α for the simulated example. Once again, the estimation of α appears to be very close to the true proportions.

3.4.3 Study of the evolution of the size on the network

In this section, we aim to present the influence of the size of the network (*i.e.* the number N of nodes) on the classification performance and on the time it takes our algorithm to calculate the estimations. To this end, we have simulated networks according to the scenario 1 with $Q = 4$ groups and the size of network varying between 100 and 400 nodes. For each size N, our VEM algorithm has been applied on 20 simulated networks and the classification performance has been assessed over the ARI Rand (1971) criterion by comparing the partition found by our algorithm with the simulated partition. the adjusted Rand index

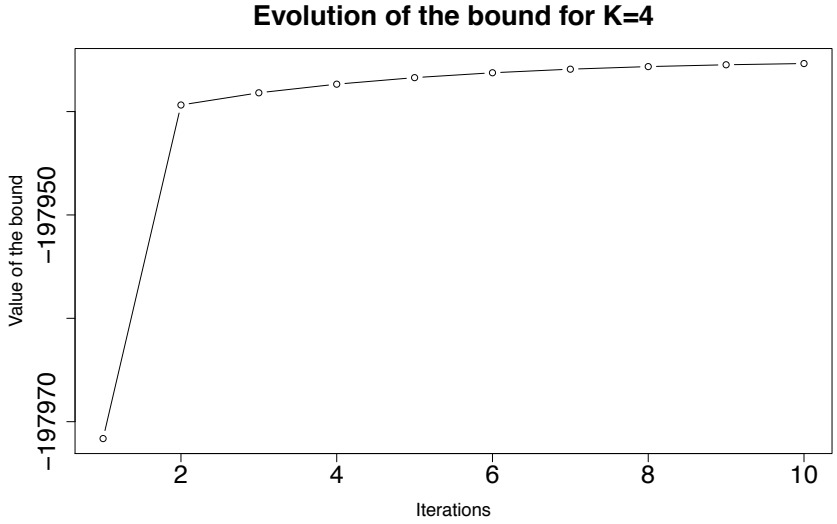


Figure 3.5: Evolution of the bound $\tilde{\mathcal{L}}$ for $Q = 4$.

(ARI), serves as a widely accepted criterion for the difficult task of clustering evaluation. The ARI looks at all pairs of nodes and checks whether they are classified in the same group or not in both partitions. As a result, an ARI value close to 1 means that the partitions are similar and, in our case, that the VEM algorithm succeeds in recovering the simulated partition.

Figure 3.7 presents the values of ARI criterion depending on the size N of network in the form of boxplots, (left) the binary case and (right) categorical case. It appears that in the binary case ($C = 1$), the classification results for $N = 100$ are satisfactory whereas, they are very good from $N = 150$ nodes onwards.

In the categorical case, the task seems to the substantially more difficult, we need to wait that networks to reach the size equivalent to $N = 250$ nodes to obtain good classification results.

Lastly, Table 3.5 shows the average execution time required by the VEM algorithm depending on the size N of the network, in the binary case (left) and the categorical case (right), with $T = 10$. On the basis of these results, we can see that our VEM algorithm gives the results with logical time considering the number of parameters estimated. In addition, we note that the changeover of the categorical case does not entail a cost-benefit where compared to the binary case.

3.4.4 Choice of Q

We now focus on the evaluation of the criterion we proposed to select the number Q of latent groups. Since our approach aims at searching the unobserved clustering partition of the nodes, we chose here to evaluate the combination of

Cluster	1	2	3	4
1	0.90	0.01	0.01	0.01
2	0.01	0.60	0.01	0.01
3	0.01	0.01	0.50	0.01
4	0.01	0.01	0.01	0.40

Actual values

Cluster	1	2	3	4
1	0.89	0.01	0.01	0.01
2	0.01	0.59	0.01	0.01
3	0.01	0.01	0.48	0.01
4	0.01	0.01	0.01	0.39

Estimated values

Table 3.3: Actual (left) and estimated (right) values for the terms Π^1_{ql} of the tensor matrix Π. See text for details.

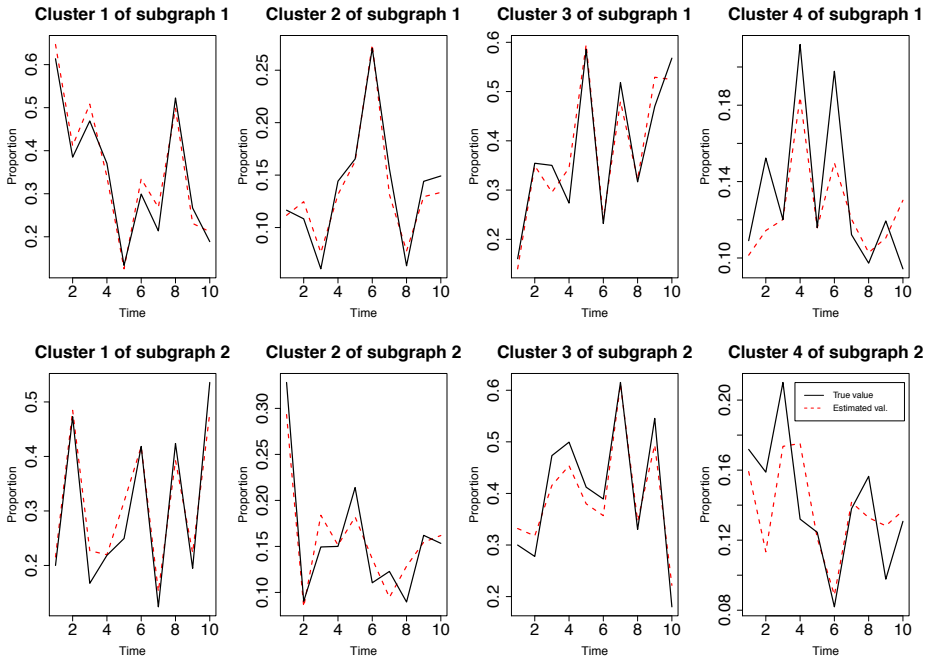


Figure 3.6: Actual (dashed red lines) and estimated (solid black lines) values of the group proportions for the simulated example ($Q = 4$ groups and $S = 2$ subgraphs).

group	1	2	3
1	0.100	0.990	0.990
2	0.990	0.400	0.990
3	0.99	0.990	0.600

Actual values of $C = 0$

group	1	2	3
1	0.109	0.990	0.989
2	0.990	0.609	0.989
3	0.989	0.989	0.407

estimated values for $C = 0$

group	1	2	3
1	0.450	0.005	0.005
2	0.005	0.300	0.005
3	0.005	0.005	0.200

Actual values of $C = 1$

group	1	2	3
1	0.442	0.004	0.005
2	0.004	0.197	0.005
3	0.005	0.005	0.294

estimated values for $C = 1$

group	1	2	3
1	0.450	0.005	0.005
2	0.005	0.300	0.005
3	0.005	0.005	0.200

Actual values of $C = 2$

group	1	2	3
1	0.448	0.004	0.004
2	0.004	0.193	0.004
3	0.004	0.005	0.297

estimated values for $C = 2$

Table 3.4: Actual (left) and estimated (right) values for the matrix Π^c_{ql} with $c \in (0, 1, 2)$ (from top to bottom).

our VEM algorithm with the BIC criterion by comparing the resulting partition with the actual one (the simulated partition). In the clustering community, the adjusted Rand index (ARI) (Rand, 1971) serves as a widely accepted criterion for the difficult task of clustering evaluation. The ARI looks at all pairs of nodes and check wether they are classified in the same group or not in both partitions. As a result, an ARI value close to 1 means that the partitions are similar and, in our case, that the VEM algorithm succeeds in recovering the simulated partition.

To validate the combination of our VEM algorithm with the BIC criterion, the analysis was repeated for 50 different data sets, generated according to scenario 2, for a number Q of latent groups ranging from 3 to 6. This allows us to both verify the consistency of the BIC criterion and to study the clustering ability of our approach. Figure 3.8 shows the repartition of the criterion values (left panel) as well as the associated ARI values (right panel). These results first confirm that BIC is a valid criterion for selecting the number of groups in this context. Indeed, the value $Q = 4$ is the one which is the most frequently associated with the highest value of BIC. We remind that $Q = 4$ is the actual number of latent groups. One can also observe that the partition resulting from our VEM algorithm is associated, for this value of Q, to an ARI value extremely close to 1 which denotes a good matching with the actual partition of the data.

3.4.5 Comparison with the other stochastic models

Our third set of experiments now aims at comparing the performance of our approach to that of state-of-the-art methods. We are here interested in the

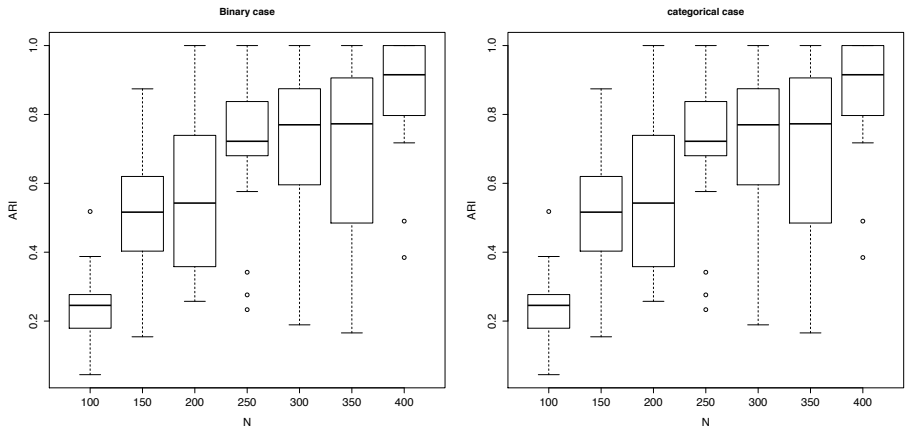


Figure 3.7: ARI values depending on the size N of the networks in the binary case (left) and categorical (right).

Size of network (N)	Time of execution ($C = 1$)	Time of execution ($C = 2$)
100	0.14 min	0.22 min
150	0.29 min	0.32 min
200	0.43 min	0.46 min
250	0.55 min	0.58 min
300	0.78 min	0.75 min
350	0.96 min	0.94 min
400	1.26 min	1.21 min

Table 3.5: The average execution time required by the VEM algorithm depending on the size N of the network, in the binary case (left) and the categorical case (right) for $T = 10$.

comparison of dRSM with the following methods: SBM (Nowicki and Snijders, 2001), RSM (Jernite et al., 2014) and dM3SBM (Ho et al., 2011). Once again, the evaluation of the results is done using the ARI criterion. In order to fit a SBM on a dynamic network, we ran the `mixer` package (Ambroise et al., 2010) for the R software at each time t and the ARI is then computed on the concatenation of all group labels. However, let us notice that SBM was not able to handle networks with categorical edges (scenario 3). For RSM, we used the `Rambo` package (Bouveyron et al., 2013) for R, on an aggregated version of the whole network. Conversely to SBM, RSM is only able to deal with categorical networks and, consequently, it works only in scenario 4. Finally, we used the Matlab toolbox `dM3SBM`, kindly provided by the authors, to fit the dM3SBM on the dynamic networks. However, dM3SBM is also not able to handle networks with categorical edges (scenario 4).

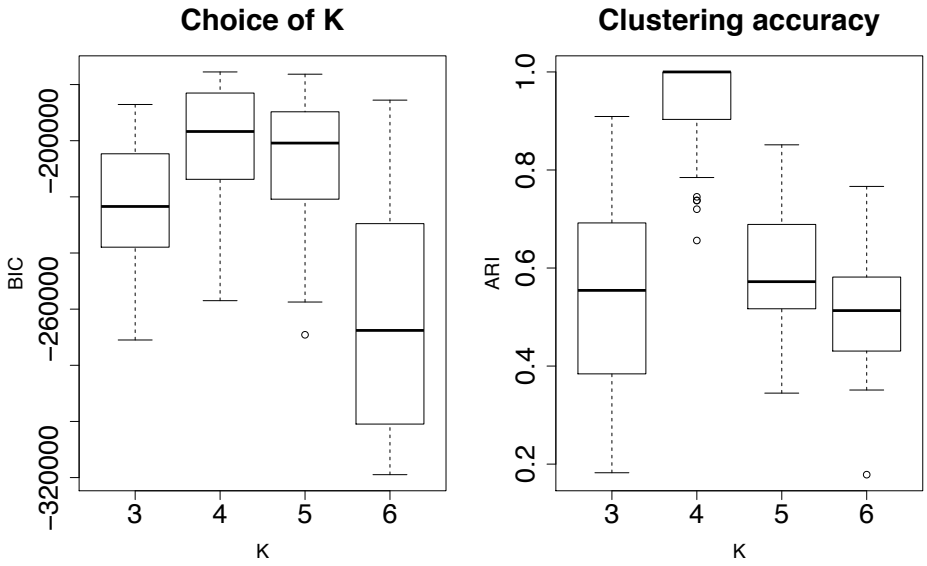


Figure 3.8: *Criterion and ARI values over 50 networks generated.*

In order to consider a wide type of networks, we compare here the methods over the five simulation scenarios. We remind that Table 3.2 summarizes the main features of each scenario. This comparison has been conducted in two different situations: with and without the knowledge of the actual number of clusters. Table 3.6 presents the clustering results for the four studied methods in the case where the actual number $Q = 4$ of groups has been provided to each method. Conversely, Table 3.7 presents the clustering results when the methods have to look for the value of Q. Reported values are averaged ARI values (with standard deviations) on 20 networks for each scenario. The average selected number Q of latent groups is also provided for Table 3.7.

First, for scenarios 0, 1 and 2, which consider dynamic networks with binary edges ($C = 1$) and with only one subgraph ($S = 1$), one can see on Tables 3.6 and 3.7 that SBM is, as expected, not able to handle the network dynamic. Indeed, SBM obtains a low ARI value in all situations, even though it correctly estimates the number of clusters (Table 3.7). Conversely, the two dynamic methods (dM3SBM and dRSM) turn out to be able to recover the clustering structure of the dynamic networks. One can however notice that dRSM significantly outperforms dM3SBM in this situation. Notice also the accurate estimation of the number Q of clusters made by dRSM (Table 3.7).

In scenario 3, the simulated dynamic networks are now made of two subgraphs ($S = 2$), still with binary edges ($C = 1$). Naturally, SBM does not perform well in this situation too. The dM3SBM provides clustering results similar to the ones of previous scenarios: it globally succeeds in recovering the dynamic but fails in recognizing the clustering pattern. On the other hand, dRSM provides again accurate clustering results associated with good estimations of Q, meaning

Method	Scenario 0	Scenario 1	Scenario 2	Scenario 3	Scenario 4
SBM	0.10±0.04	0.12±0.05	0.18±0.07	0.14±0.09	–
RSM	–	–	–	–	0.01±0.01
dM3SBM	0.36±0.09	0.30±0.16	0.25±0.16	0.32±0.20	–
dRSM	1.00±0.00	0.98±0.04	0.90±0.20	0.97±0.07	0.75±0.24

Table 3.6: Clustering results for the four studied methods on networks simulated according to the five scenarios. The actual number $Q = 4$ of groups has been provided to each method here. Average ARI values are reported (with standard deviations) and results are averaged on 20 networks for each scenario.

that it succeeds in identifying both the dynamic and clustering patterns.

Finally, scenario 4 considers the case of dynamic networks with two subgraphs ($S = 2$) and categorical edges ($C = 2$). Only RSM and dRSM are able to deal with this kind of networks. Similarly to SBM in previous scenarios, RSM does not succeed in recovering the dynamic and provides very unsatisfactory clustering results. Conversely, dRSM gives very good clustering results regarding the difficulty of the situation. It is worth noticing the sharp estimation made by dRSM of the number Q of group in this case too. This confirms the efficiency of both our inference algorithm and our model selection criterion.

We also used scenario 4 to highlight that providing the methodology with the right subgraph structure helps in clustering the vertices. Thus, with the knowledge of the actual number of clusters, we ran dRSM with the wrong subgraph structure ($S = 1$), and we obtained an average ARI of 0.54±0.2. This result is to be compared to the ARI performances for scenario 4, as presented in Table 3.6.

Method	Scenario 0		Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	ARI	Q	ARI	Q	ARI	Q	ARI	Q	ARI	Q
SBM	0.01±0.04	4.00±0.00	0.18±0.13	3.94±0.71	0.21±0.11	3.97±0.46	0.13±0.05	4.16±0.79	–	–
RSM	–	–	–	–	–	–	–	–	0.01±0.01	2.00±0.00
dM3SBM	0.01±0.01	5.55±1.39	0.35±0.21	5.95±1.15	0.30±0.21	4.35±1.63	0.32±0.19	5.15±1.17	–	–
dRSM	1.00±0.00	4.00±0.00	0.87±0.17	4.01±0.65	0.89±0.21	4.10±0.30	0.85±0.22	4.10±0.45	0.68±0.30	4.05±0.51

Table 3.7: Clustering results for the four studied methods on networks simulated according to the five scenarios. Average ARI values are reported (with standard deviations) as well as the selected number Q of latent groups. Results are averaged on 20 networks for each scenario.

3.5 Conclusion

This chapter has considered the problem of analyzing dynamic networks with categorical edges and for which a subgraph partition is known. This kind of networks is frequent in a wide range of scientific fields, such as Geography in particular. For this purpose, we proposed an extension of the RSM model to the dynamic setting. The new model, called dRSM, uses a state space model to model the evolution of the latent group proportions over time. A variational expectation maximization (VEM) algorithm is proposed to perform inference. We have shown in particular that the variational approximations lead to a new state space model from which the parameters can be estimated using the standard Kalman filter and the Rauch-Tung-Striebel (RTS) smoother. Model selection is also considered through an approximate BIC criterion.

Numerical experiments have highlighted the main features of the dRSM model and have demonstrated the efficiency of both the VEM algorithm and the model selection criterion. A numerical comparison has also shown that existing methods, dynamic or not, are less flexible and efficient than dRSM when applied to dynamic networks.

CHAPTER 4

STOCHASTIC TOPIC BLOCK MODEL

Contents

Due to the significant increase of communications between individuals via social media (Facebook, Twitter, Linkedin) or electronic formats (email, web, e-publication) in the past two decades, network analysis has become a unavoidable discipline. Many random graph models have been proposed to extract information

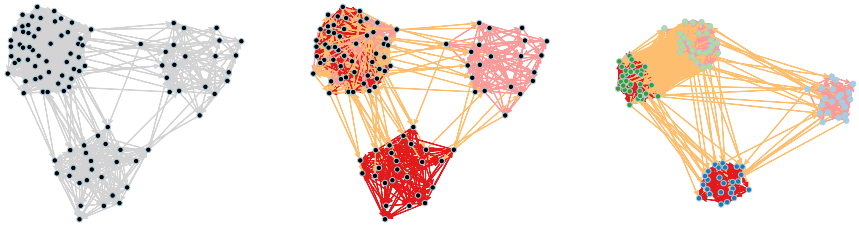


Figure 4.1: A sample network made of 3 “communities” where one of the communities is made of two topic-specific groups. The left panel only shows the observed (binary) edges in the network. The center panel shows the network with only the partition of edges into 3 topics (edge colors indicate the majority topics of texts). The right panel shows the network with the clustering of its nodes (vertex colors indicate the groups) and the majority topic of the edges. The latter visualization allows to see the topic-conditional structure of one of the three communities.

from networks based on person-to-person links only, without taking into account information on the contents. This paper introduces the stochastic topic block model (STBM), a probabilistic model for networks with textual edges. We address here the problem of discovering meaningful clusters of vertices that are coherent from both the network interactions and the text contents. A classification variational expectation-maximization (C-VEM) algorithm is proposed to perform inference. Simulated data sets are considered in order to assess the proposed approach and to highlight its main features.

4.1 Introduction

Ranging from communication to co-authorship networks, it is nowadays particularly frequent to observe networks with textual edges. It is obviously of strong interest to be able to model and cluster the vertices of those networks using information on both the network structure and the text contents. Techniques able to provide such a clustering would allow a deeper understanding of the studied networks. As a motivating example, Figure 4.1 shows a network made of 3 "communities" of vertices where one of the communities can in fact be split into two separate groups based on the topics of communication between nodes of these groups (see legend of Figure 4.1 for details). Despite the important efforts in both network analysis and text analytics, only a few works have focused on the joint modeling of network vertices and textual edges.

In the context of statistical models for the joint analysis of texts and networks,

a few recent works have focused on the joint modeling of texts and networks. Those works are mainly motivated by the will of analyzing social networks, such as Twitter or Facebook, or electronic communication networks. Some of them are partially based on latent Dirichlet allocation (LDA) (Section 2.5.2): the author-topic (AT) (Steyvers et al., 2004; Rosen-Zvi et al., 2004) and the author-recipient-topic (ART) (McCallum et al., 2005) models. The AT model extends LDA to include authorship information whereas the ART model includes authorships and information about the recipients. Though potentially powerful, these models do not take into account the network structure (communities, stars, ...) while the concept of community is very important in the context of social networks, in the sense that a community is a group of users sharing similar interests. Among the most advanced models for the joint analysis of texts and networks, the first models which explicitly take into account both text contents and network structure are the community-user-topic (CUT) models proposed by (Zhou et al., 2006). Two models are proposed: CUT1 and CUT2, which differ on the way they construct the communities. Indeed, CUT1 determines the communities only based on the network structure whereas CUT2 model the communities based on the content information solely. The CUT models therefore deal each with only a part of the problem we are interested in. It is also worth noticing that the authors of these models rely for inference on Gibbs sampling which may prohibit their use on large networks. A second attempt was made by Pathak et al. (2008) who extended the ART model by introducing the community-author-recipient-topic (CART) model. The CART model adds to the ART model that authors and recipients belong to latent communities and allows CART to recover groups of nodes that are homogenous both regarding the network structure and the message contents. Notice that CART allows the nodes to be part of multiple communities and each couple of actors to have a specific topic. Thus, though extremely flexible, CART is also a highly parametrized model. In addition, the recommended inference procedure based on Gibbs sampling may also prohibit its application to large networks. More recently, the topic-link LDA (Liu et al., 2009) also performs topic modeling and author community discovery in a unified framework. As its name suggests, topic-link LDA extends LDA with a community layer where the link between two documents (and consequently its authors) depends on both topic proportions and author latent features through a logistic transformation. However, whereas CART focuses only on directed networks, topic-link LDA is only able to deal with undirected networks. On the positive side, the authors derive a variational EM algorithm for inference, allowing topic-link LDA to eventually be applied to large networks. Finally, a family of 4 topic-user-community models (TUCM) were proposed by Sachan et al. (2012). The TUCM models are designed such that they can find topic-meaningful communities in networks with different types of edges. This in particular relevant in social networks such as Twitter where different types of interactions (followers, tweet, re-tweet, ...) exist. Another specificity of the TUCM models is that they allow both multiple community and topic memberships. Inference is also done here through Gibbs sampling, implying a possible scale limitation.

Contributions of the present chapter We propose here a new generative model for the clustering of networks with textual edges, such as communication or co-authorship networks. Conversely to existing works which have either too simple or highly-parametrized models for the network structure, our model relies for the network modeling on the SBM model which offers a sufficient flexibility with a reasonable complexity. This model is one of the few able to recover different topological structures such as communities, stars or disassortative clusters (see Latouche et al., 2012, for instance). Regarding the topic modeling, our approacg is based on the LDA model, in which the topics are conditioned on the latent groups. Thus, the proposed modeling will be able to exhibit node partitions that are meaningful both regarding the network structure and the topics, with a model of limited complexity, highly interpretable, and for both directed and undirected networks. In addition, the proposed inference procedure – a classification-VEM algorithm – allows the use of our model on large-scale networks.

In this chapter we proposed stochastic topic block model (STBM) which is introduced in Section 4.2. The model inference is discussed in Section 4.3 as well as model selection. Section 4.4 is devoted to numerical experiments highlighting the main features of the proposed approach and proving the validity of the inference procedure. Section 4.5 finally provides some concluding remarks.

4.2 The model

This section presents the notations used in the paper and introduces the STBM model. The joint distributions of the model to create edges and the corresponding documents are also given.

4.2.1 Context and notations

A directed network with M vertices, described by its $M \times M$ adjacency matrix X, is considered. Thus, $X_{ij} = 1$ if there is an edge from vertex i to vertex j, 0 otherwise. The network is assumed not to have any self-loop and therefore $X_{ii} = 0$ for all i. If an edge from i to j is present, then it is characterized by a set of D_{ij} documents, denoted $W_{ij} = (W_{ij}^d)_d$. Each document W_{ij}^d is made of a collection of N_{ij}^d words $W_{ij}^d = (W_{ij}^{dn})_n$. In the directed scenario considered, W_{ij} can model for instance a set of emails or text messages sent from actor i to actor j. Note that all the methodology proposed in this paper easily extends to undirected networks. In such a case, $X_{ij} = X_{ji}$ and $W_{ij}^d = W_{ji}^d$ for all i and j. The set W_{ij}^d of documents can then model for example books or scientific papers written by both i and j. In the following, we denote $W = (W_{ij})_{ij}$ the set of all documents exchanged, for all the edges present in the network.

Our goal is to cluster the vertices into Q latent groups sharing homogeneous connection profiles, *i.e.* find an estimate of the set $Z = (Z_1, \ldots, Z_M)$ of latent

variables Z_i such that $Z_{iq} = 1$ if vertex i belongs to cluster q, and 0 otherwise. Although in some cases, discrete or continuous edges are taken into account, the literature on networks focuses on modeling the presence of edges as binary variables. The clustering task then consists in building groups of vertices having similar trends to connect to others. In this paper, the connection profiles are both characterized by the presence of edges and the documents between pairs of vertices. Therefore, we aim at uncovering clusters by integrating these two sources of information. Two nodes in the same cluster should have the same trend to connect to others, and when connected, the documents they are involved in should be made of words related to similar topics.

4.2.2 Modeling the presence of edges

In order to model the presence of edges between pairs of vertices, a stochastic block model (Wang and Wong, 1987; Nowicki and Snijders, 2001) is considered. Thus, the vertices are assumed to be spread into Q latent clusters such that $Z_{iq} = 1$ if vertex i belongs to cluster q, and 0 otherwise. In practice, the binary vector Z_i is assumed to be drawn from a multinomial distribution

$$Z_i \sim \mathcal{M}\left(1, \rho = (\rho_1, \ldots, \rho_Q)\right),$$

where ρ denotes the vector of class proportions. By construction, $\sum_{q=1}^{Q} \rho_q = 1$ and $\sum_{q=1}^{Q} Z_{iq} = 1, \forall i$.

An edge from i to j is then sampled from a Bernoulli distribution, depending on their respective clusters

$$X_{ij} | Z_{iq} Z_{jr} = 1 \sim \mathcal{B}(\pi_{qr}). \tag{4.1}$$

In words, if i is in cluster q and j in r, then X_{ij} is 1 with probability π_{qr}. In the following, we denote π the $Q \times Q$ matrix of connection probabilities. Note that in the undirected case, π is symmetric.

All vectors Z_i are sampled independently, and given $Z = (Z_1, \ldots, Z_M)$, all edges in X are assumed to be independent. This leads to the following joint distribution

$$p(X, Z | \rho, \pi) = p(X | Z, \pi) p(Z | \rho),$$

where

$$\begin{aligned} p(X | Z, \pi) &= \prod_{i \neq j}^{M} p(X_{ij} | Z_i, Z_j, \pi) \\ &= \prod_{i \neq j}^{M} \prod_{q,l}^{Q} p(X_{ij} | \pi_{qr})^{Z_{iq} Z_{jr}}, \end{aligned}$$

and

$$p(Z|\rho) = \prod_{i=1}^{M} p(Z_i|\rho)$$
$$= \prod_{i=1}^{M}\prod_{q=1}^{Q} \rho_q^{Z_{iq}}.$$

4.2.3 Modeling the construction of documents

As mentioned previously, if an edge is present from vertex i to vertex j, then a set of documents $W_{ij} = (W_{ij}^d)_d$, characterizing the oriented pair (i, j), is assumed to be given. Thus, in a generative perspective, the edges in X are first sampled using previous section. Given X, the documents in $W = (W_{ij})_{ij}$ are then constructed. The generative process we consider to build documents is strongly related to the latent Dirichlet allocation (LDA) model of Blei et al. (2003). The link between STBM and LDA is made clear in the following section. The STBM model relies on two concepts at the core of the SBM and LDA models respectively. On the one hand, a generalization of the SBM model would assume that any kind of relationships between two vertices can be explained by their latent clusters only. In the LDA model on the other hand, the main assumption is that words in documents are drawn from a mixture distribution over topics, each document d having its own vector of topic proportions θ_d. The STBM model combines these two concepts to introduce a new generative procedure for documents in networks.

Each pair of clusters (q, r) of vertices is first associated to a vector of topic proportions $\theta_{qr} = (\theta_{qrk})_k$ sampled independently from a Dirichlet distribution

$$\theta_{qr} \sim \mathrm{Dir}\left(\alpha = (\alpha_1, \ldots, \alpha_K)\right),$$

such that $\sum_{k=1}^{K} \theta_{qrk} = 1, \forall (q, r)$. We denote $\theta = (\theta_{qr})_{qr}$. The nth word W_{ij}^{dn} of documents d in W_{ij} is then associated to a latent topic vector Y_{ij}^{dn} assumed to be drawn from a multinomial distribution, depending on the latent vectors Z_i and Z_j

$$Y_{ij}^{dn} | \left\{Z_{iq} Z_{jr} X_{ij} = 1, \theta\right\} \sim \mathcal{M}\left(1, \theta_{qr} = (\theta_{qr1}, \ldots, \theta_{qrK})\right). \tag{4.2}$$

Note that $\sum_{k=1}^{K} Y_{ij}^{dnk} = 1, \forall (i, j, d), X_{ij} = 1$. Equations (4.1) and (4.2) are related: they both involve the construction of random variables depending on the cluster assignment of vertices i and j. Thus, if an edge is present ($X_{ij} = 1$) and if i is in cluster q and j in r, then the word W_{ij}^{dn} is in topic k ($Y_{ij}^{dnk} = 1$) with probability θ_{qrk}.

Then, given Y_{ij}^{dn}, the word W_{ij}^{dn} is assumed to be drawn from a multinomial distribution

$$W_{ij}^{dn} | Y_{ij}^{dnk} = 1 \sim \mathcal{M}\left(1, \beta_k = (\beta_{k1}, \ldots, \beta_{kV})\right), \tag{4.3}$$

where V is the number of (different) words in the vocabulary considered and $\sum_{v=1}^{V} \beta_{kv} = 1, \forall k$ as well as $\sum_{v=1}^{V} W_{ij}^{dnv} = 1, \forall (i,j,d,n)$. Therefore, if W_{ij}^{dn} is from topic k, then it is associated to word v of the vocabulary ($W_{ij}^{dnv} = 1$) with probability β_{kv}. Equations (4.2) and (4.3) lead to the following mixture model for words over topics

$$W_{ij}^{dn} | \{Z_{iq} Z_{jr} X_{ij} = 1, \theta\} \sim \sum_{k=1}^{K} \theta_{qrk} \mathcal{M}(1, \beta_k),$$

where the $K \times V$ matrix $\beta = (\beta_{kv})_{kv}$ of probabilities does not depend on the cluster assignments. Note that words of different documents d and $d^{'}$ in W_{ij} have the same mixture distribution which only depends on the respective clusters of i and j. We also point out that words of the vocabulary appear in any document d of W_{ij} with probabilities

$$\mathbb{P}(W_{ij}^{dnv} = 1 | Z_{iq} Z_{jr} X_{ij} = 1, \theta) = \sum_{k=1}^{K} \theta_{qrk} \beta_{kv}.$$

Because pairs (q,r) of clusters can have different vectors of topics proportions θ_{qr}, the documents they are associated with can have different mixture distribution of words over topics. For instance, most words exchanged from vertices of cluster q to vertices of cluster r can be related to *mathematics* while vertices from q' can discuss with vertices of r' with words related to *cinema* and in some cases to *sport*.

All the latent variables Y_{ij}^{dn} are assumed to be sampled independently and, given the latent variables, the words W_{ij}^{dn} are assumed to be independent. Denoting $Y = (Y_{ij}^{dn})_{ijdn}$, this leads to the following joint distribution

$$p(W, Y, \theta | X, Y, \beta) = p(W | X, Y, \beta) p(Y | X, Z, \theta) p(\theta),$$

where

$$\begin{aligned} p(W | X, Y, \beta) &= \prod_{i \neq j}^{M} \left\{ \prod_{d=1}^{D_{ij}} \prod_{n=1}^{N_{ij}^d} p(W_{ij}^{dn} | Y_{ij}^{dn}, \beta) \right\}^{X_{ij}} \\ &= \prod_{i \neq j}^{M} \left\{ \prod_{d=1}^{D_{ij}} \prod_{n=1}^{N_{ij}^d} \prod_{k=1}^{K} p(W_{ij}^{dn} | \beta_k)^{Y_{ij}^{dnk}} \right\}^{X_{ij}}, \end{aligned}$$

and

$$\begin{aligned} p(Y | X, Z, \theta) &= \prod_{i \neq j}^{M} \left\{ \prod_{d=1}^{D_{ij}} \prod_{n=1}^{N_{ij}^d} p(Y_{ij}^{dn} | Z_i, Z_j, \theta) \right\}^{X_{ij}} \\ &= \prod_{i \neq j}^{M} \left\{ \prod_{d=1}^{D_{ij}} \prod_{n=1}^{N_{ij}^d} \prod_{q,r}^{Q} p(Y_{ij}^{dn} | \theta_{qr})^{Z_{iq} Z_{jr}} \right\}^{X_{ij}}, \end{aligned}$$

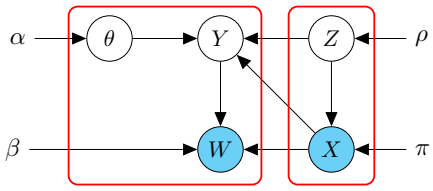


Figure 4.2: Graphical representation of the stochastic topic block model.

as well as

$$p(\theta) = \prod_{q,r}^{Q} \mathrm{Dir}(\theta_{qr}; \alpha).$$

4.2.4 Link with LDA and SBM

The full joint distribution of the STBM model is given by

$$p(X, W, Z, Y, \theta|\rho, \pi, \beta) = p(W, Y, \theta|X, Z, \beta)p(X, Z|\rho, \pi), \tag{4.4}$$

and the corresponding graphical model is provided in Figure 4.2. Thus, all the documents in W are involved in the full joint distribution through $p(W, Y, \theta|X, Z, \beta)$. Now, let us assume that Z is available. It then possible to reorganize the documents in W such that $W = (\tilde{W}_{qr})_{qr}$ where $\tilde{W}_{qr} = \left\{W_{ij}^{d}, \forall(d, i, j), Z_{iq}Z_{jr}X_{ij} = 1\right\}$ is the set of all documents exchanged from any vertex i in cluster q to any vertex j in cluster r. As mentioned in the previous section, each word W_{ij}^{dn} has a mixture distribution over topics which only depends on the clusters of i and j. Because all words in $\tilde{W}_{qr}$ are associated with the same pair (q, r) of clusters, they share the same mixture distribution. Removing temporarily the knowledge of (q, r), *i.e.* simply seeing $\tilde{W}_{qr}$ as a document d, the sampling scheme described previously then corresponds to the one of a LDA model with $D = Q^2$ independent documents $\tilde{W}_{qr}$, each document having its own vector θ_{qr} of topic proportions. The model is then characterized by the matrix β of probabilities. Note that contrary to the original LDA model (Blei et al., 2003), the Dirichlet distributions considered for the θ_{qr} depend on a fixed vector α.

As mentioned in Section 4.2.2, the second part of Equation (4.4) involves the sampling of the clusters and the construction of binary variables describing the presence of edges between pairs of vertices. Interestingly, it corresponds exactly to the complete data likelihood of the SBM model, as considered in Zanghi et al. (2008) for instance. Such a likelihood term only involves the model parameters ρ and π.

4.3 Inference

We aim at maximizing the complete data log-likelihood

$$\log p(X, W, Z|\rho, \pi, \beta) = \log \sum_Y \int_\theta p(X, W, Z, Y, \theta|\rho, \pi, \beta) d\theta, \tag{4.5}$$

with respect to the model parameters (ρ, π, β) and the set $Z = (Z_1, \ldots, Z_M)$ of cluster membership vectors. Note that Z is not seen here as a set of latent variables over which the log-likelihood should be integrated out, as in standard expectation maximization (EM) (Dempster et al., 1977) or variational EM algorithms (Hathaway, 1986). Moreover, the goal is not to provide any approximate posterior distribution of Z given the data and model parameters. Conversely, Z is seen here as a set of (binary) vectors for which we aim at providing estimates. This choice is motivated by the key property of the STBM model, *i.e.* for a given Z, the full joint distribution factorizes into a LDA like term and SBM like term. In particular, given Z, words in W can be seen as being drawn from a LDA model with $D = Q^2$ documents (see Section 4.2.4), for which fast optimization tools have been derived, as pointed out in the introduction. Note that the choice of optimizing a complete data log-likelihood with respect to the set of cluster membership vectors has been considered in the literature, for simple mixture model such as Gaussian mixture models, but also for the SBM model (Zanghi et al., 2008). The corresponding algorithm, so called classification EM (CEM) (Celeux and Govaert, 1991) alternates between the estimation of Z and the estimation of the model parameters.

As mentioned previously, we introduce our methodology in the directed case. However, we emphasize that the STBM package for R we developed, implements the inference strategy for both directed and undirected networks.

4.3.1 Variational decomposition

Unfortunately, in our case, Equation (4.5) is not tractable. Moreover the posterior distribution $p(Y, \theta|X, W, Z, \rho, \pi, \beta)$ does not have any analytical form. Therefore, following the work of Blei et al. (2003) on the LDA model, we propose to rely on a variational decomposition. In the case of the STBM model, it leads to

$$\log p(X, W, Z|\rho, \pi, \beta) = \mathcal{L}\left(R(\cdot); Z, \rho, \pi, \beta\right) + \mathrm{KL}\left(R(\cdot) \parallel p(\cdot|X, W, Z, \rho, \pi, \beta)\right),$$

where

$$\mathcal{L}\left(R(\cdot); Z, \rho, \pi, \beta\right) = \sum_Y \int_\theta R(Y, \theta) \log \frac{p(X, W, Y, Z, \theta|\rho, \pi, \beta)}{R(Y, \theta)} d\theta, \tag{4.6}$$

and KL denotes the Kullback-Leibler divergence between the true and approximate posterior distributions of (Y, θ), given the data and model parameters

$$\mathrm{KL}\left(R(\cdot) \parallel p(\cdot|X, W, Z, \rho, \pi, \beta)\right) = -\sum_Y \int_\theta R(Y, \theta) \log \frac{p(Y, \theta|X, W, Z, \rho, \pi, \beta)}{R(Y, \theta)} d\theta.$$

Since $\log p(X, W, Z|\rho, \pi, \beta)$ does not depend on the distribution $R(Y, \theta)$, maximizing the lower bound $\mathcal{L}$ with respect to $R(Y, \theta)$ induces a minimization of the KL divergence. As in Blei et al. (2003), we assume that $R(Y, \theta)$ can be factorized over the latent variables in θ and Y. In our case, this translates into

$$R(Y, \theta) = R(Y)R(\theta) = R(\theta) \prod_{i \neq j, X_{ij}=1}^{M} \prod_{d=1}^{D_{ij}} \prod_{n=1}^{N_{ij}^d} R(Y_{ij}^{dn}).$$

4.3.2 Model decomposition

As pointed out in Section 4.2.4, the set of latent variables in Z allows the decomposition of the full joint distribution in two terms, from the sampling of Z and X to the construction of documents given X and Z. When deriving the lower bound (4.6), this property leads to

$$\mathcal{L}\left(R(\cdot); Z, \rho, \pi, \beta\right) = \tilde{\mathcal{L}}\left(R(\cdot); Z, \beta\right) + \log p(X, Z|\rho, \pi),$$

where

$$\tilde{\mathcal{L}}\left(R(\cdot); Z, \beta\right) = \sum_{Y} \int_{\theta} R(Y, \theta) \log \frac{p(W, Y, \theta|X, Z, \beta)}{R(Y, \theta)} d\theta, \tag{4.7}$$

and $\log p(X, Z|\rho, \pi)$ is the complete data log-likelihood of the SBM model. The parameter β and the distribution $R(Y, \theta)$ are only involved in the lower bound $\tilde{\mathcal{L}}$ while ρ and π only appear in $\log p(X, Z|\rho, \pi)$. Therefore, given Z, these two terms can be maximized independently. Moreover, given Z, $\tilde{\mathcal{L}}$ is the lower bound for the LDA model, as proposed by Blei et al. (2003), after building the set $W = (\tilde{W}_{qr})_{qr}$ of $D = Q^2$ documents, as described in Section 4.2.4. In the next section, we derive a VEM algorithm to maximize $\tilde{\mathcal{L}}$ with respect β and $R(Y, \theta)$, which essentially corresponds to the VEM algorithm of Blei et al. (2003). Then, $\log p(X, Z|\rho, \pi)$ is maximized with respect to ρ and π to provide estimates. Finally, $\mathcal{L}\left(R(\cdot); Z, \rho, \pi, \beta\right)$ is maximized with respect to Z, which is the only term involved in both $\tilde{\mathcal{L}}$ and the SBM complete data log-likelihood. Because the methodology we propose requires a variational EM approach as well as a classification step, to provide estimates of Z, we call the corresponding strategy a classification VEM (C-VEM) algorithm.

4.3.3 Optimization

In this section, we derive the optimization steps of the C-VEM algorithm we propose, which aims at maximizing the lower bound $\mathcal{L}$. The algorithm alternates between the optimization of $R(Y, \theta)$, Z and (ρ, π, β) until convergence of the lower bound.

Estimation of $R(Y, \theta)$ The following propositions give the update formulate of the E step of the VEM algorithm applied on Equation (4.7).

Proposition 4.3.1 *The VEM update step for each distribution $R(Y_{ij}^{dn})$ is given by*

$$R(Y_{ij}^{dn}) = \mathcal{M}\left(Y_{ij}^{dn}; 1, \phi_{ij}^{dn} = (\phi_{ij}^{dn1}, \ldots, \phi_{ij}^{dnK}\right),$$

where

$$\phi_{ij}^{dnk} \propto \left(\sum_{v=1}^{V} W_{ij}^{dnv} \log \beta_{kv}\right) \prod_{q,r}^{Q} \left(\psi(\gamma_{qrk} - \psi(\sum_{l=1}^{K} \gamma_{qrl})\right)^{Z_{iq}Z_{jr}}, \forall(d,n,k).$$

ϕ_{ij}^{dnk} is the (approximate) posterior distribution of words W_{ij}^{dn} being in topic k.

***Proof of Prop. 4.3.1*:** The VEM update step for each distribution $R(Y_{ij}^{dn})$, $X_{ij} = 1$, is given by

$$\begin{aligned}
\log R(Y_{ij}^{dn}) &= \mathrm{E}_{Y^{\setminus i,j,d,n},\theta}[\log p(W|X,Y,\beta) + \log p(Y|X,Z,\theta)] + \text{const} \\
&= \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^{dn}} \sum_{k=1}^{K} Y_{ij}^{dnk} \sum_{v=1}^{V} W_{ij}^{dnv} \log \beta_{kv} + \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^{dn}} \sum_{q,r}^{Q} Z_{iq}Z_{jr} \sum_{k=1}^{K} Y_{ij}^{dnk} \mathrm{E}_{\theta_{qr}}[\log \theta_{qrk}] + \text{const} \\
&= \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^{dn}} \sum_{k=1}^{K} Y_{ij}^{dnk} \left(\sum_{v=1}^{V} W_{ij}^{dnv} \log \beta_{kv} + \sum_{q,r}^{Q} Z_{iq}Z_{jr}\Big(\psi(\gamma_{qrk} - \psi(\sum_{k=1}^{K} \gamma_{qrk})\Big)\right) + \text{const},
\end{aligned} \tag{4.8}$$

where all terms that do not depend on Y_{ij}^{dn} have been put into the constant term const. Moreover, $\psi(\cdot)$ denotes the digamma function. The functional form of a multinomial distribution is then recognized in (4.8)

$$R(Y_{ij}^{dn}) = \mathcal{M}\left(Y_{ij}^{dn}; 1, \phi_{ij}^{dn} = (\phi_{ij}^{dn1}, \ldots, \phi_{ij}^{dnK}\right),$$

where

$$\phi_{ij}^{dnk} \propto \left(\sum_{v=1}^{V} W_{ij}^{dnv} \log \beta_{kv}\right) \prod_{q,r}^{Q} \left(\psi(\gamma_{qrk} - \psi(\sum_{l=1}^{K} \gamma_{qrl})\right)^{Z_{iq}Z_{jr}}.$$

ϕ_{ij}^{dnk} is the (approximate) posterior distribution of words W_{ij}^{dn} being in topic k. ∎

Proposition 4.3.2 *The VEM update step for distribution $R(\theta)$ is given by*

$$R(\theta) = \prod_{q,r}^{Q} \mathrm{Dir}(\theta_{qr}; \gamma_{qr} = (\gamma_{qr1}, \ldots, \gamma_{qrK})),$$

where

$$\gamma_{qrk} = \alpha_k + \sum_{i \neq j}^{M} X_{ij} Z_{iq} Z_{jr} \sum_{d=1}^{N_{ij}^{d}} \sum_{n=1}^{N_{ij}^{dn}} \phi_{ij}^{dnk}, \forall(q,r,k).$$

***Proof of Prop. 4.3.2*:** The VEM update step for distribution $R(\theta)$ is given by

$$\begin{aligned}\log R(\theta) &= \mathrm{E}_Y[\log p(Y|X,Z,\theta)] + \text{const} \\ &= \sum_{i\neq j}^{M} X_{ij} \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^d} \sum_{q,r}^{Q} Z_{iq} Z_{jr} \sum_{k=1}^{K} \mathrm{E}_{Y_{ij}^{dn}}[Y_{ij}^{dnk}] \log \theta_{qrk} + \sum_{q,r}^{Q} \sum_{k=1}^{K} (\alpha_k - 1) \log \theta_{qrk} + \text{const} \\ &= \sum_{q,r}^{Q} \sum_{k=1}^{K} \left(\alpha_k + \sum_{i\neq j}^{M} X_{ij} Z_{iq} Z_{jr} \sum_{d=1}^{N_{ij}^d} \sum_{n=1}^{N_{ij}^{dn}} \phi_{ij}^{dnk} - 1 \right) + \text{const.}\end{aligned}$$

We recognize the functional form of a product of Dirichlet distributions

$$R(\theta) = \prod_{q,r}^{Q} \mathrm{Dir}(\theta_{qr}; \gamma_{qr} = (\gamma_{qr1}, \ldots, \gamma_{qrK})),$$

where

$$\gamma_{qrk} = \alpha_k + \sum_{i\neq j}^{M} X_{ij} Z_{iq} Z_{jr} \sum_{d=1}^{N_{ij}^d} \sum_{n=1}^{N_{ij}^{dn}} \phi_{ij}^{dnk}.$$

∎

4.3.4 Derivation of the lower bound $\tilde{\mathcal{L}}(R(\cdot); Z, \beta)$

Estimation of the model parameters Maximizing the lower bound $\mathcal{L}$ in Equation (4.7) is used to provide estimates of the model parameters (ρ, π, β). We recall that β is only involved in $\tilde{\mathcal{L}}$ while (ρ, π) only appear in the SBM complete

data log-likelihood. The derivation of $\tilde{\mathcal{L}}$:

$$\begin{aligned}
\tilde{\mathcal{L}}\left(R(\cdot); Z, \beta\right) &= \sum_{Y} \int_{\theta} R(Y,\theta) \log \frac{p(W, Y, \theta | X, Z, \beta)}{R(Y,\theta)} d\theta \\
&= \mathrm{E}_Y[\log p(W|X,Y,\beta)] + \mathrm{E}_{Y,\theta}[\log p(Y|X,Z,\theta)] + \mathrm{E}_\theta[\log p(\theta)] - \mathrm{E}_Y[\log R(Y)] - \mathrm{E}_\theta[\log R(\theta)] \\
&= \sum_{i \neq j}^{M} X_{ij} \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^{dn}} \sum_{k=1}^{K} \phi_{ij}^{dnk} \sum_{v=1}^{V} W_{ij}^{dnv} \log \beta_{kv} \\
&\quad + \sum_{i \neq j}^{M} X_{ij} \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^{dn}} \sum_{q,r}^{Q} Z_{iq} Z_{jr} \sum_{k=1}^{K} \phi_{ij}^{dnk} \Big(\psi(\gamma_{qrk}) - \psi(\sum_{l=1}^{K} \gamma_{qrl}) \Big) \\
&\quad + \sum_{q,r}^{Q} \left(\log \Gamma(\sum_{l=1}^{K} \alpha_k) - \sum_{l=1}^{K} \log \Gamma(\alpha_l) + \sum_{k=1}^{K} (\alpha_k - 1) \Big(\psi(\gamma_{qrk}) - \psi(\sum_{l=1}^{K} \gamma_{qrl}) \Big) \right) \\
&\quad - \sum_{i \neq j}^{M} X_{ij} \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^{dn}} \sum_{k=1}^{K} \phi_{ij}^{dnk} \log \phi_{ij}^{dnk} \\
&\quad - \sum_{q,r}^{Q} \left(\log \Gamma(\sum_{l=1}^{K} \gamma_{qrl}) - \sum_{l=1}^{K} \log \Gamma(\gamma_{qrl}) + \sum_{k=1}^{K} (\gamma_{qrk} - 1) \Big(\psi(\gamma_{qrk}) - \psi(\sum_{l=1}^{K} \gamma_{qrl}) \Big) \right)
\end{aligned} \tag{4.9}$$

Proposition 4.3.3 *The estimates of β, ρ, and π, are given by*

$$\beta_{kv} \propto \sum_{i \neq j}^{M} X_{ij} \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^{dn}} \phi_{ij}^{dnk} W_{ij}^{dnv}, \forall (k, v),$$

$$\rho_q \propto \sum_{i=1}^{Q} Z_{iq}, \forall q,$$

$$\pi_{qr} = \frac{\sum_{i \neq j}^{M} \sum_{q,r}^{Q} Z_{iq} Z_{jr} X_{ij}}{\sum_{i \neq j}^{M} \sum_{q,r}^{Q} Z_{iq} Z_{jr}}, \forall (q, r).$$

Proof of Prop. 4.3.3:

Optimization of β: In order to maximize the lower bound $\tilde{\mathcal{L}}\left(R(\cdot); Z, \beta\right)$, we isolate the terms in (4.9) that depend on β and add Lagrange multipliers to satisfy the constraints $\sum_{v=1}^{V} \beta_{kv} = 1, \forall k$

$$\tilde{\mathcal{L}}_\beta = \sum_{i \neq j}^{M} X_{ij} \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^{dn}} \sum_{k=1}^{K} \phi_{ij}^{dnk} \sum_{v=1}^{V} W_{ij}^{dnv} \log \beta_{kv} + \sum_{k=1}^{K} \lambda_k (\sum_{v=1}^{V} \beta_{kv} - 1).$$

Setting the derivative, with respect to β_{kv}, to zero, we find

$$\beta_{kv} \propto \sum_{i \neq j}^{M} X_{ij} \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^{dn}} \phi_{ij}^{dnk} W_{ij}^{dnv}.$$

Optimization of ρ: Only the distribution $p(Z|\rho)$ in the complete data log-likelihood $\log p(X, Z|\rho, \pi)$ depends on the parameter vector ρ of cluster proportions. Taking the log and adding a Lagrange multiplier to satisfy the constraint $\sum_{q=1}^{Q} \rho_q = 1$, we have

$$\log p(Z|\rho) + \sum_{i=1}^{M} \sum_{q=1}^{Q} Z_{iq} \log \rho_q.$$

Taking the derivative with respect ρ to zero, we find

$$\rho_q \propto \sum_{i=1}^{Q} Z_{iq}.$$

Optimization of π: Only the distribution $p(X|Z, \pi)$ in the complete data log-likelihood $\log p(X, Z|\rho, \pi)$ depends on the parameter matrix π of connection probabilities. Taking the log we have

$$\log p(X|Z, \pi) + \sum_{i \neq j}^{M} \sum_{q,r}^{Q} Z_{iq} Z_{jr} \Big(X_{ij} \log \pi_{qr} + (1 - X_{ij}) \log(1 - \pi_{qr}) \Big)$$

Taking the derivative with respect to π_{qr} to zero, we obtain

$$\pi_{qr} = \frac{\sum_{i \neq j}^{M} \sum_{q,r}^{Q} Z_{iq} Z_{jr} X_{ij}}{\sum_{i \neq j}^{M} \sum_{q,r}^{Q} Z_{iq} Z_{jr}}.$$

∎

Estimation of Z At this step, the model parameters (ρ, π, β) along with the distribution $R(Y, \theta)$ are held fixed. Therefore, the lower bound $\mathcal{L}$ in (4.7) only involves the set Z of cluster membership vectors. Looking for the optimal solution Z maximizing this bound is not feasible since it involves testing the Q^M possible cluster assignments. However, heuristics are available to provide local maxima for this combinatorial problem. These so called *greedy* methods have been used for instance to look for communities in networks by Newman (2004); Blondel et al. (2008) but also for the SBM model (Côme and Latouche, 2015). They are sometimes referred to as *on line* clustering methods (Zanghi et al., 2008).

The algorithm cycles randomly through the vertices. At each step, a single vertex is considered and all membership vectors Z_j are held fixed, except Z_i.

If i is currently in cluster q, then the method looks for every possible label swap, *i.e.* removing i from cluster q and assigning it to a cluster $r \neq q$. The corresponding change in the SBM complete data log-likelihood is then computed. If no label swap induces an increase in the SBM complete data log-likelihood, then Z_i remains unchanged. Otherwise, the label swap that yields the maximal increase is applied, and Z_i is changed accordingly.

4.3.5 Initialization strategy and model selection

The C-VEM introduced in the previous section allows the estimation of $R(Y, \theta)$, Z, as well as (ρ, π, β), for a fixed number Q of clusters and a fixed number K of topics. As any EM-like algorithms, the C-VEM method depends on the initialization and is only guaranteed to converge to a local optimum (Bilmes, 1998). Strategies to tackle this issue include simulated annealing and the use of multiple initializations (Biernacki et al., 2003). In this work, we choose the latter option. Our C-VEM algorithm is run for several initializations of a k-means like algorithm on a distance matrix between the vertices obtained as follows.

1. The VEM algorithm (Blei et al., 2003) for LDA is applied on the aggregation of all documents exchanged from vertex i to vertex j, for each pair (i, j) of vertices, in order to characterize a type of interaction from i to j. Thus, a $M \times M$ matrix A is first built such that $A_{ij} = k$ if k is the majority topic used by i when discussing with j.

2. The distance $M \times M$ matrix Δ is then computed as follows

$$\Delta(i, j) = \sum_{h=1}^{N} \delta(A_{ih} \neq A_{jh}) X_{ih} X_{jh} + \sum_{h=1}^{N} \delta(A_{hi} \neq A_{hj}) X_{hi} X_{hj}. \quad (4.10)$$

   The first term looks at all possible edges from i and j towards a third vertex h. If both i and j are connected to h, *i.e.* $X_{ih} X_{jh} = 1$, the edge types A_{ih} and A_{jh} are compared. By symmetry, the second term looks at all possible edges from a vertex h to both i as well as j, and compare their types. Thus, the distance computes the number of discordances in the way both i and j connect to other vertices or vertices connect to them.

Regarding model selection, since a model based approach is proposed here, two STBM models will be seen as different is they have different values of Q and/or K. Therefore, the task of estimating Q and K can be viewed as a model selection problem. Many model selection criteria have been proposed in the literature, such as the Akaike information criterion (Akaike, 1973) (AIC) and the Bayesian information criterion (Schwarz, 1978) (BIC). In this paper, because the optimization procedure considered involves the optimization of the binary matrix Z, we rely on a ICL-like criterion. This criterion was originally proposed by Biernacki et al. (2000a) for Gaussian mixture models. In the STBM context, it aims at approximating the integrated complete data log-likelihood $\log p(X, W, Z)$.

Proposition 4.3.4 *Relying on two Laplace approximations, a variational estimation, and Stirling formula, a ICL criterion for the STBM model can be obtained*

$$ICL_{STBM} = BIC_{LDA|Z} + ICL_{SBM},$$

where

$$BIC_{LDA|Z} = \tilde{\mathcal{L}}(R(\cdot); Z, \beta) - \frac{K(V-1)}{2} \log Q^2,$$

is the BIC criterion for the LDA model with the Q^2 documents $\tilde{W}_{qr}$ built from Z (see Section 4.2.4), and

$$ICL_{SBM} = \max_{\rho,\pi} \log p(X, Z|\rho, \pi, Q) - \frac{Q^2}{2} \log M(M-1) - \frac{Q-1}{2} \log M$$

is the ICL criterion of the SBM model, as introduced by Daudin et al. (2008).

Proof of Prop. 4.3.4:

Assuming that the prior distribution over the model parameters (ρ, π, β) can be factorized, the integrated complete data log-likelihood $\log p(X, W, Z|K, Q)$ is given by

$$\begin{aligned}
\log p(X, W, Z|K, Q) &= \log \int_{\rho,\pi,\beta} p(X, W, Z, \rho, \pi, \beta|K, Q) d\rho d\pi d\theta \\
&= \log \int_{\rho,\pi,\beta} p(X, W, Z|\rho, \pi, \beta, K, Q) p(\rho|Q) p(\pi|Q) p(\beta|K) d\rho d\pi d\beta.
\end{aligned}$$

Note that the dependency on K and Q is made explicit here, in all expressions. In all other sections of this chapter, we did not include these terms to keep the notations uncluttered. We find

$$\begin{aligned}
\log p(X, W, Z|K, Q) &= \log \int_{\rho,\pi,\beta} \left(\sum_Y \int_\theta p(X, W, Y, Z, \theta|\rho, \pi, \beta, K, Q) d\theta \right) p(\rho|Q) p(\pi|Q) p(\beta|K) d\rho d\pi d\beta \\
&= \log \int_{\rho,\pi,\beta} \left(\sum_Y \int_\theta p(W, Y, \theta|X, Z, \beta, K, Q) p(X, Z|\rho, \pi, Q) d\theta \right) p(\rho|Q) p(\pi|Q) p(\beta|K) d\rho d\pi d\beta \\
&= \log \int_{\rho,\pi,\beta} p(W|X, Z, \beta, K, Q) p(X|Z, \pi, Q) p(Z|\rho, Q) p(\rho|Q) p(\pi|Q) p(\beta|K) d\rho d\pi d\beta \\
&= \log \int_\beta p(W|X, Z, \beta, K, Q) p(\beta|K) d\beta + \log \int_\pi p(X|Z, \pi, Q) p(\pi|Q) d\pi \\
&\quad + \log \int_\rho p(Z|\rho, Q) p(\rho|Q) d\rho.
\end{aligned} \tag{4.11}$$

Following the derivation of the ICL criterion, we apply a Laplace (BIC-like) approximation on the second term of Equation (4.11). Moreover, considering a Jeffreys prior distribution for ρ and using Stirling formula for large values of M, we obtain

$$\log \int_\pi p(X|Z, \pi, Q) p(\pi|Q) d\pi \approx \max_\pi \log p(X|Z, \pi, Q) - \frac{Q^2}{2} \log M(M-1),$$

as well as

$$\log \int_{\rho} p(Z|\rho, Q)p(\rho|Q)d\rho \approx \max_{\rho} \log p(Y|\rho, Q) - \frac{Q-1}{2} \log M.$$

For more details, we refer to Biernacki et al. (2000a). Furthermore, we emphasize that adding these two approximations leads to the ICL criterion for the SBM model, as derived by Daudin et al. (2008)

$$\begin{aligned} ICL_{SBM} &= \max_{\pi} \log p(X|Z, \pi, Q) - \frac{Q^2}{2} \log M(M-1) + \max_{\rho} \log p(Z|\rho, Q) - \frac{Q-1}{2} \log M \\ &= \max_{\rho,\pi} \log p(X, Z|\rho, \pi, Q) - \frac{Q^2}{2} \log M(M-1) - \frac{Q-1}{2} \log M. \end{aligned}$$

In Daudin et al. (2008), $M(M-1)$ is replaced by $M(M-1)/2$ and Q^2 by $Q(Q+1)/2$ since they considered undirected networks.

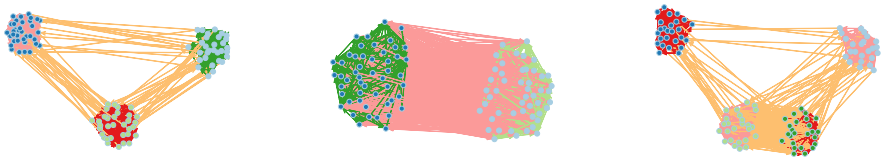
Now, it is worth taking a closer look at the first term of Equation (4.11). This term involves a marginalization over β. Let us emphasize that $p(W|X, Z, \beta, K, Q)$ is related to the LDA model and involves a marginalization over θ (and Y). Because we aim at approximating the first term of Equation (4.11), also with a Laplace (BIC-like) approximation, it is crucial to identify the number of observations in the associated likelihood term $p(W|X, Z, \beta, K, Q)$. As pointed out in Section 4.2.4, given Z (and θ), it is possible to reorganize the documents in W as $W = (\tilde{W}_{qr})_{qr}$ is such a way that all words in $\tilde{W}_{qr}$ follow the same mixture distribution over topics. Each aggregated document $\tilde{W}_{qr}$ has its own vector θ_{qr} of topic proportions and since the distribution over θ factorizes ($p(\theta) = \prod_{q,r}^{Q} p(\theta_{qr})$), we find

$$\begin{aligned} p(W|X, Z, \beta, K, Q) &= \int_{\theta} p(W|X, Z, \theta, \beta, K, Q)p(\theta|K, Q)d\theta \\ &= \prod_{q,r}^{Q} \int_{\theta_{qr}} p(\tilde{W}_{qr}|\theta_{qr}, \beta, K, Q)p(\theta_{qr}|K)d\theta_{qr} \\ &= \prod_{q,r}^{Q} \ell(\tilde{W}_{qr}|\beta, K, Q), \end{aligned}$$

where $\ell(\tilde{W}_{qr}|\beta, K, Q)$ is exactly the likelihood term of the LDA model associated with document $\tilde{W}_{qr}$, as described in Blei et al. (2003). Thus

$$\log \int_{\beta} p(W|X, Z, \beta, K, Q)p(\beta|K)d\beta = \log \int_{\beta} p(\beta|K) \prod_{q,r}^{Q} \ell(\tilde{W}_{qr}|\beta, K, Q)d\beta. \tag{4.12}$$

Applying a Laplace approximation on Equation (4.12) is then equivalent to deriving a BIC-like criterion for the LDA model with documents in $W = (\tilde{W}_{qr})_{qr}$. In the LDA model, the number of observations in the penalization term of BIC



Scenario A Scenario B Scenario C

Figure 4.3: Networks sampled according to the three simulation scenarios A, B and C. See text for details.

is the number of documents. In our case, this leads to

$$\log \int_{\beta} p(W|X,Z,\beta,K,Q)p(\beta|K)d\beta \approx \max_{\beta} \log p(W|X,Z,\beta,K,Q) - \frac{K(V-1)}{2}\log Q^2. \tag{4.13}$$

Unfortunately, $\log p(W|X,Z,\beta,K,Q)$ is not tractable and so we propose to replace it with its variational approximation $\tilde{\mathcal{L}}$, after convergence of the C-VEM algorithm. By analogy with ICL_{SBM}, we call the corresponding criterion $BIC_{LDA|Z}$ such that

$$\log p(X,W,Z|K,Q) \approx BIC_{LDA|Z} + ICL_{SBM}. \quad \blacksquare$$

4.4 Numerical experiments

This section aims at highlighting the main features of the proposed approach on synthetic data and at proving the validity of the inference algorithm presented in the previous section. Model selection is also considered to validate the criterion choice. Numerical comparisons with state-of-the-art methods conclude this section.

4.4.1 Experimental setup

First, regarding the parametrization of our approach, we chose $\alpha_k = 1, \forall k$ which induces a uniform distribution over the topic proportions θ_{qr}.

Second, regarding the simulation setup and in order to illustrate the interest of the proposed methodology, three different simulation setups will be used in this section. To simplify the characterization and facilitate the reproducibility of the experiments, we designed three different scenarios. They are as follows:

- scenario A consists in networks with $Q = 3$ groups, corresponding to clear communities, where persons within a group talk preferentially about a

Scenario	A	B	C
M (nb of nodes)	100		
K (topics)	4	3	3
Q (groups)	3	2	4
ρ (group prop.)	$(1/Q, ..., 1/Q)$		
π (connection prob.)	$\begin{cases} \pi_{qq} = & 0.25 \\ \pi_{qr,\, r \neq q} = & 0.01 \end{cases}$	$\pi_{qr,\, \forall q,r} = 0.25$	$\begin{cases} \pi_{qq} = & 0.25 \\ \pi_{qr,\, r \neq q} = & 0.01 \end{cases}$
θ (prop. of topics)	$\begin{cases} \theta_{111} = \theta_{222} = & 1 \\ \theta_{333} = & 1 \\ \theta_{qr4,\, r \neq q} = & 1 \\ \text{otherwise} & 0 \end{cases}$	$\begin{cases} \theta_{111} = \theta_{222} = & 1 \\ \theta_{qr3,\, r \neq q} = & 1 \\ \text{otherwise} & 0 \end{cases}$	$\begin{cases} \theta_{111} = \theta_{331} = & 1 \\ \theta_{222} = \theta_{442} = & 1 \\ \theta_{qr3,\, r \neq q} = & 1 \\ \text{otherwise} & 0 \end{cases}$

Table 4.1: Parameter values for the three simulation scenarios (see text for details).

unique topic and use a different topic when talking with persons of other groups. Thus, those networks contain $K = 4$ topics.

- scenario B consists in networks with a unique community where the $Q = 2$ groups are only differentiated by the way they discuss within and between groups. Persons within groups #1 and #2 talk preferentially about topics #1 and #2 respectively. A third topic is used for the communications between persons of different groups.
- scenario C, finally, consists in networks with $Q = 4$ groups which use $K = 3$ topics to communicate. Among the 4 groups, two groups correspond to clear communities where persons talk preferentially about a unique topic within the communities. The two other groups correspond to a single community and are only discriminated by the topic used in the communications. People from group #3 use topic #1 and the topic #2 is used in group #4. The third topic is used for communications between groups.

For all scenarios, the simulated messages are sampled from four texts from BBC news: one text is about the birth of Princess Charlotte, the second one is about black holes in astrophysics, the third one is focused on UK politics and the last one is about cancer diseases in medicine. All messages are made of 150 words. Table 4.1 provides the parameter values for the three simulation scenarios. Figure 4.3 shows simulated networks according to the three simulation scenarios. It is worth noticing that all simulation scenarios have been designed such that they do not to strictly follow the STBM model and therefore they do not favor the model we propose in comparisons.

4.4.2 Introductory example

As an introductory example, we consider a network of $M = 100$ nodes sampled according to scenario C (3 communities, $Q = 4$ groups and $K = 3$ topics). This scenario corresponds to a situation where both network structure and topic information are needed to correctly recover the data structure. Indeed, groups #3 and #4 form a single community when looking at the network structure and

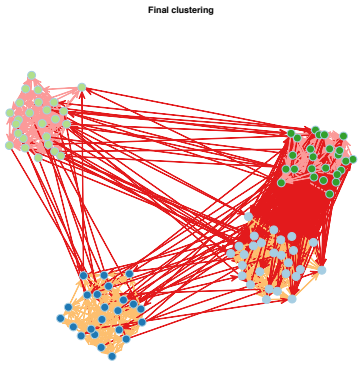


Figure 4.4: Clustering result for the introductory example (scenario C). See text for details.

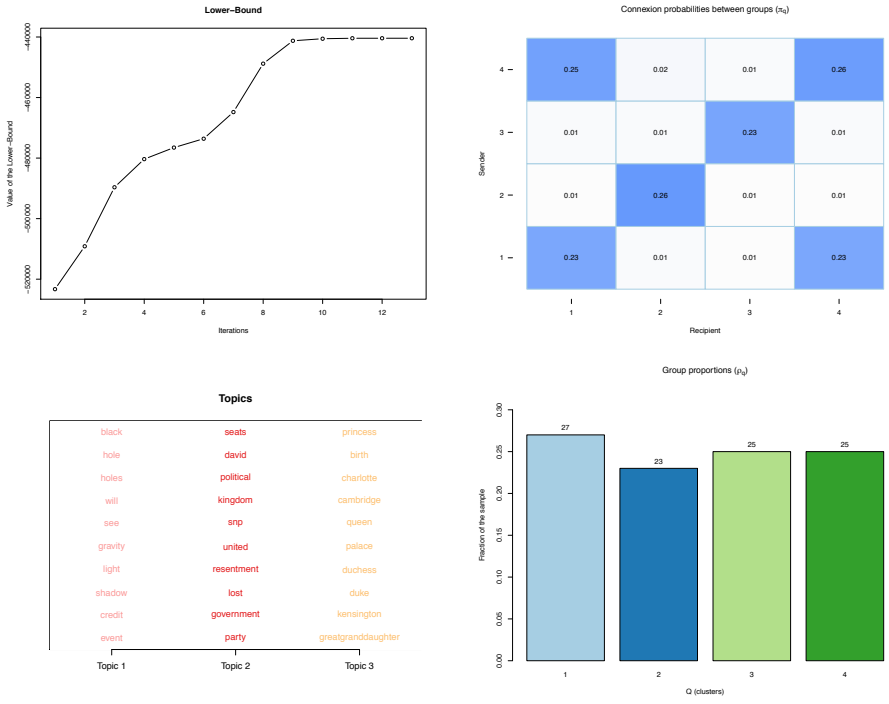


Figure 4.5: Clustering result for the introductory example (scenario C). See text for details.

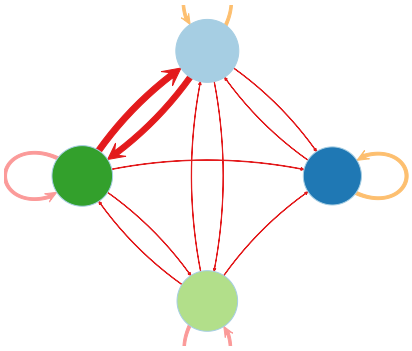


Figure 4.6: Introductory example: summary of connexion probabilities between groups (π, edge widths), group proportions (ρ, node sizes) and most probable topics for group interactions (edge colors).

it is necessary to look at the way they communicate to discriminate the two groups.

The C-VEM algorithm for STBM was run on the network with the actual number of groups and topics (the problem of model selection will be considered in next section). Figure 4.4 first shows the obtained clustering, which is here perfect both regarding the simulated node and edges partitions. More interestingly, Figure 4.5 allows to visualize the evolution of the lower bound $\mathcal{L}$ along the algorithm iterations (top-left panel), the estimated model parameters π and ρ (right panels) and the most frequent words in the 3 found topics (left-bottom panel). It turns out that both the model parameters, π and ρ (see Table 4.1 for actual values), and the topic meanings are well recovered. STBM indeed perfectly recover the three themes that we used for simulating the textual edges: one is a "royal baby" topic, one is a political one and the last on is focused on Physics. Notice also that this result was obtained in only a few iterations of the C-VEM algorithm, that we proposed for inferring STBM models.

A useful and compact view of both parametersπ and ρ, and of the most probable topics for group interactions can be offered by Figure 4.6. Here, edge widths correspond to connexion probabilities between groups (π), the node sizes are proportional to group proportions (ρ) and edge colors indicate the majority topics for group interactions. It is important to notice that, even though only the most probable topic is displayed here, each textual edge may use different topics.

4.4.3 Model selection

This experiment focuses on the ability of the ICL criterion to select the most appropriate values for Q and K. To this end, we simulated 50 networks according

Scenario A ($Q = 3$, $K = 4$)						
$K \backslash Q$	1	2	3	4	5	6
1	0	0	0	0	0	0
2	12	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	**82**	2	0	2
5	0	0	2	0	0	0
6	0	0	0	0	0	0

Scenario B ($Q = 2$, $K = 3$)						
$K \backslash Q$	1	2	3	4	5	6
1	0	0	0	0	0	0
2	12	0	0	0	0	0
3	0	**88**	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Scenario C ($Q = 4$, $K = 3$)						
$K \backslash Q$	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	2	**82**	0	0
4	0	0	0	16	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Table 4.2: Percentage of selections by ICL for each STBM model (Q, K) on 50 simulated networks of each of three scenarios. Highlighted rows and columns correspond to the actual values for Q and K.

to each of the three scenarios and STBM was applied on those networks for values of Q and K ranging from 1 to 6. Table 4.2 presents the percentage of selections by ICL for each STBM model (Q, K) on 50 simulated networks of each of three scenarios.

In the three different situations, ICL succeeds most of the time to identify the actual combination of the number of groups and topics. For scenarios A and B, when ICL does not select the correct values for Q and K, the criterion seems to underestimate the values of Q and K whereas it tends to overestimate them in case of scenario C. One can also notice that wrongly selected models are usually close to the simulated one. Let us also recall that, since the data are not strictly simulated according to a STBM model, the ICL criterion does not have the model which generated the data in the set of tested models. This experiment allows to validate ICL as a model selection tool for STBM.

4.4.4 Benchmark study

This third experiment aims at comparing the ability of STBM to recover the network structure both in term of node partition and topics. STBM is here compared to SBM, using the mixer package (Ambroise et al., 2010), and LDA, using the topicmodels package (Grun and Hornik, 2013). Obviously, SBM and LDA will be only able to recover either the node partition or the topics. We chose here to evaluate the results by comparing the resulting node and topic partitions with the actual ones (the simulated partitions). In the clustering

		Scenario A		Scenario B		Scenario C	
Easy	Method	node ARI	edge ARI	node ARI	edge ARI	node ARI	edge ARI
	SBM	1.00±0.00	–	0.01±0.01	–	0.69±0.07	–
	LDA	–	0.97±0.06	–	1.00±0.00	–	1.00±0.00
	STBM	0.98±0.04	0.98±0.04	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00

		Scenario A		Scenario B		Scenario C	
Hard 1	Method	node ARI	edge ARI	node ARI	edge ARI	node ARI	edge ARI
	SBM	0.01±0.01	–	0.01±0.01	–	0.01±0.01	–
	LDA	–	0.90±0.17	–	1.00±0.00	–	0.99±0.01
	STBM	1.00±0.00	0.90±0.13	1.00±0.00	1.00±0.00	1.00±0.00	0.98±0.03

		Scenario A		Scenario B		Scenario C	
Hard 2	Method	node ARI	edge ARI	node ARI	edge ARI	node ARI	edge ARI
	SBM	1.00±0.00	–	-0.01±0.01	–	0.65±0.05	–
	LDA	–	0.21±0.13	–	0.08±0.06	–	0.09±0.05
	STBM	0.99±0.02	0.99±0.01	0.59±0.35	0.54±0.40	0.68±0.07	0.62±0.14

Table 4.3: Clustering results for the SBM, LDA and STBM on 20 networks simulated according to the three scenarios. Average ARI values are reported with standard deviations for both node and edge clustering. The "Easy" situation corresponds to the simulation situation describes in Table 4.1. In the "Hard 1" situation, the communities are very few differentiated ($\pi_{qq} = 0.25$ and $\pi_{q\neq r} = 0.2$, except for scenario B). The "Hard 2" situation finally corresponds to a setup where 40% of message words are sampled in different topics than the actual topic.

community, the adjusted Rand index (ARI) (Rand, 1971) serves as a widely accepted criterion for the difficult task of clustering evaluation. The ARI looks at all pairs of nodes and checks whether they are classified in the same group or not in both partitions. As a result, an ARI value close to 1 means that the partitions are similar.

In addition to the different simulation scenarios, we considered three different situations: the standard simulation situation as described in Table 4.1 (hereafter "Easy"), a simulation situation (hereafter "Hard 1") where the communities are less differentiated ($\pi_{qq} = 0.25$ and $\pi_{q\neq r} = 0.2$, except for scenario B) and a situation (hereafter "Hard 2") where 40% of message words are sampled in different topics than the actual topic.

In the "Easy" situation, the results are coherent with our initial guess when building the simulation scenarios. Indeed, besides the fact that SBM and LDA are only able to recover one of the two partitions, scenario A is a easy situation for all methods since the clusters perfectly match the topic partition. Scenario B, which has no communities and groups only depend on topics, is obviously a difficult situation for SBM but does not disturb LDA which perfectly recovers the topics. In scenario C, LDA still succeeds in identifying the topics whereas SBM well recognize the two communities but fails in discriminating the two

groups hidden in a single community. Here, STBM obtains in all scenarios the best performance on both nodes and edges.

The "Hard 1" situation considers the case where the communities are actually not well differentiated. Here, LDA is few affected (only in scenario A) whereas SBM is no longer able to distinguish the groups of nodes. Conversely, STBM relies on the found topics to correctly identifies the node groups and obtains, here again, excellent ARI values in all the three scenarios.

The last situation, the so-called "Hard 2" case, aims to highlight the effect of the word sampling in the recovering of the used topics. On the one hand, SBM now achieves a satisfying classification of nodes for scenarios A and C while LDA fails in recovering the majority topic used for simulation. On those two scenarios, STBM performs well on both nodes and topics. This proves that STBM is also able to recover the topics in a noisy situation by relying on the network structure. On the other hand, scenario B presents an extremely difficult situation where topics are noised and there are no communities. Here, although both LDA and SBM fail, STBM achieves a satisfying result on both nodes and edges. This is, once again, an illustration of the fact that the joint modeling of network structure and topics allows to recover complex hidden structures in a network with textual edges.

4.5 Conclusion

This chapter has introduced a probabilistic model, named the stochastic topic bloc model (STBM), for the modeling and clustering of networks with textual edges. The proposed model allows the modeling of both directed and undirected networks, authorizing its application to networks of various types (communication, social medias, co-authorship, ...). A classification variational EM (C-VEM) algorithm has been proposed for model inference and model selection is done through the ICL criterion. Numerical experiments on simulated data sets have proved the effectiveness of both the model and its inference procedure.

CHAPTER 5

APPLICATIONS

Contents

In order to assess the validity of models defined in previous chapters, we apply in this chapter our algorithms to real data sets. To this end, we present, in Section 5.1, three applications of dRSM to two real-world networks. The first application looks at electronic communications between employees in the Enron company. The second one describes maritime flows around the world over the past century. These two data sets were chosen because of the existence of a temporal dimension factor; each contains an important event which allows us to evaluate the capacity of our algorithm to detect the presence of such events over time. Similarly, in Section 5.2, we also present two applications to real-world

networks: Enron emails and the NIPS conference co-authorship network. These two data sets were selected because they contain a large number of texts, enabling us once again to evaluate our algorithm in two types of networks: directed and undirected.

5.1 Applications of dRSM

In this section, we aim to apply our dRSM model, along with the corresponding VEM algorithm, to two real-world networks: the Enron email network and a maritime network. This choice was made for specific reasons. First, a significant evolution in connections between nodes has been observed over time in these networks. Second, in each, the occurrence of an important event led to great changes over time in the network structure.

We see therefore the utility of applying models which deal with dynamic networks to these data sets, in order to try to tap latent information over time. We note also that the two networks differ in size: one is large, whereas the other one is of average size.

5.1.1 Maritime flows

Maritime flows are extremely rich in real events, including information which can completely change the structure of connections between maritime ports around the world over time. Our results, obtained using the dRSM model, involves two precise situations. The first one looks at the general behavior of maritime flows for 17 successive periods from 1890 to 2008. The second, focuses on a specific event which occurred in a specific 17 year period. The example chosen was the collapse of the USSR in 1991, to examine changes which occurred during this shock.

The global maritime network

We begin by presenting an application of the proposed methodology to the analysis of a global network of maritime flows in which a temporal dynamic exists. The dynamic network was provided by Dr. César Ducruet, from the Géographie-Cités lab, who is interested in studying the evolution of maritime flows over time (www.world-seastems.cnrs.fr). The data was extracted from the well-known Lloyd's[1] list, which has recorded almost all ship movements worldwide since 1890.

Data and study protocol

Data was obtained from the printed Lloyd's voyage record published every October from 1890 to 2008. The list gives, for each merchant vessel, its successive movements from one port to another. From the raw database of vessel flows,

[1] https://www.lloydslist.com/

Time point	Date
t_1	October 1890
$t_2 \dots t_4$	October 1925 to October 1940, every five years
t_5	October 1946
t_6	October 1951
t_7	October 1960
$t_8 \dots t_{16}$	October 1965 to October 2000, every five years
t_{17}	October 2008

Table 5.1: Time points considered in the maritime network.

we extracted a dynamic network with 17 time points. The first observation is October 1890, and the final network corresponds to October 2008. Table 5.2 provides the link between the 17 time points and the actual dates.

At each time point, the adjacency matrix between ports was constructed as follows. First, for every pair of ports, we calculated the total number of ship movements between them. Then, we set the associated entry in the adjacency matrix to 1 if the number of ship movements between the two ports was greater than or equal to 1, and to 0 otherwise. The original network contained 4472 ports worldwide. We however had to reduce the network size to only 286 ports, since most of the ports were not active throughout the whole study period.

We thus applied dRSM to a maritime network showing the ship movements between 286 ports at 17 time points. Note that the study period includes many major historical and economic events (two world wars, the oil crisis, economic crises, etc.), which potentially directly affected navigation movements at a global scale, as well as affect port behaviors.

We provide a partition of the network into subgraphs, based on port's memberships of the four main maritime basins: Asia – Pacific, Europe – Atlantic, Mediterranean – Black Sea, and Middle East – Indian Ocean. Figure 5.1 presents this partition, with colors indicating the subgraphs.

To summarize, the network is an undirected and binary one without self loops, i.e., $C = 1$ and $X_{ij}^t = 1$ if port i and port j exchange at least one ship during the period t, and 0 otherwise, with $t \in \{1, \dots, 17\}$ and $S = 4$. Figure 5.2 shows the adjacency matrices, in 1890 and 2008, between the 286 ports, organized by subgraph.

Results

We used the variational EM algorithm introduced in the section 3.3.2 in order to find the latent groups that may be hidden in the data. The choice of the number of groups is made by applying the VEM algorithm for $K = 3, ..., 8$ and by then computing the associated BIC values. The retained value for K is the one associated with the highest BIC value. To ensure a good accuracy of the results, the VEM algorithm was run 5 times for each value of K. Figure 5.3 shows the evolution of the BIC criterion according to K. One can observe that

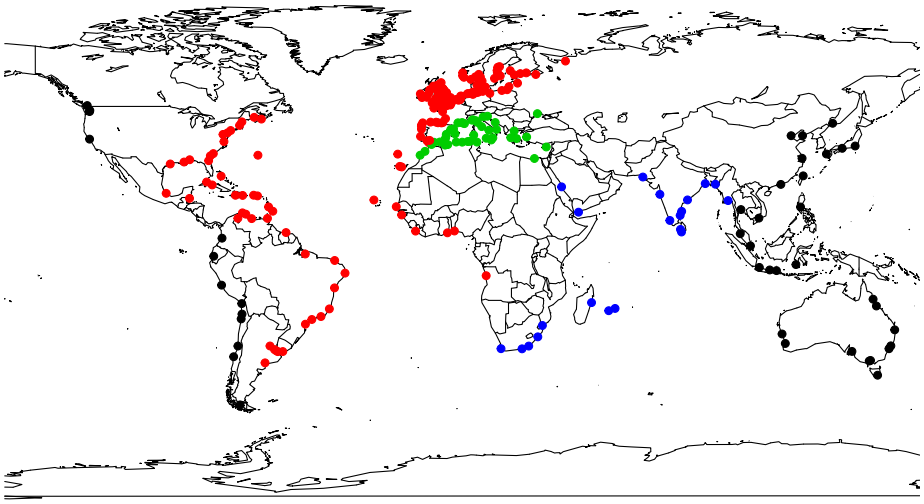


Figure 5.1: The given partition of the 286 nodes (ports) into 4 subgraphs.

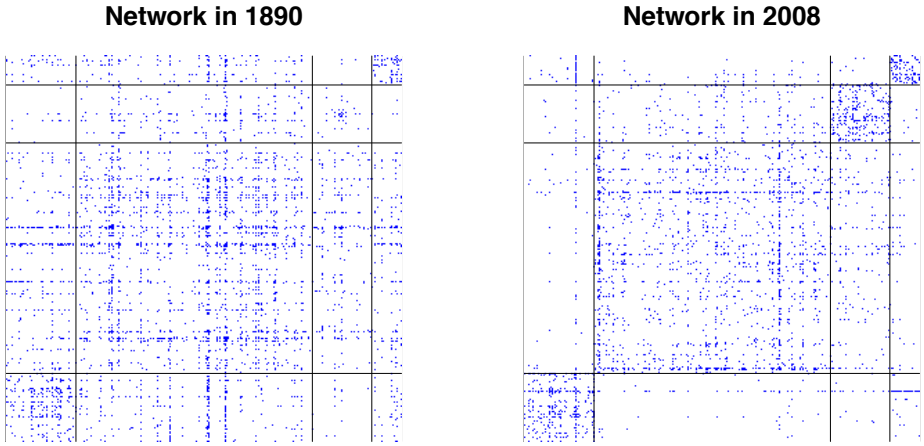


Figure 5.2: Adjacency matrix of the maritime network organized by subgraph (basin) in 1890 (left) and 2008 (right).

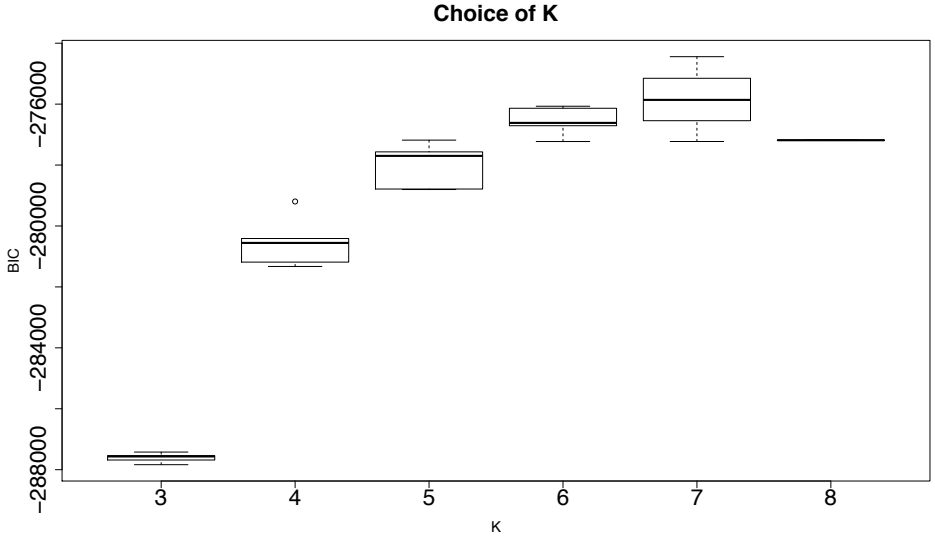


Figure 5.3: BIC values according to the number K of groups for the maritime network.

BIC peaks at $K = 7$, meaning that 7 latent groups seem to organize the network. We therefore chose this specific value for K and retained the best run for $K = 7$ over the five runs as the final clustering result.

First, it is of main interest to look at the estimated tensor matrix Π in order to understand and characterize the found latent groups. Indeed, the tensor matrix Π describes the connection probabilities between the groups and allows to figure out the different connection patterns. Since the network considered here is binary, it is enough to look at the terms Π^1_{kl} since $\Pi^0_{kl} + \Pi^1_{kl} = 1$, for all k, l. Figure 5.4 presents those estimated values. From the figure, clusters 6 and 7 appear to be groups of hubs for which the connection probabilities are large within and between clusters.

Second, the estimated group proportions over time should allow to understand the dynamic of the network. Figure 5.5 presents the evolution of those proportions over time for each subgraph. One can first observe that the proportion of cluster 6 is low and rather stable over time. This confirms that cluster 6 is a group of a limited number of hubs with a high connectivity and probably a high level of traffic. Cluster 6 includes ports such as Anvers, Rotterdam or Singapore. It is also interesting to see that, in subgraph 2 (Europe – Atlantic), the number of hubs increased until 1930, was then perturbed during the second world war and finally decreased from 1951. Conversely, in subgraph 1 (Asia – Pacific), the proportion of hubs was low until 1975 and then significantly increased. From a global point of view, one can also observe a clear and recent reorganization of the network in which hubs tend to be less numerous worldwide (and probably bigger).

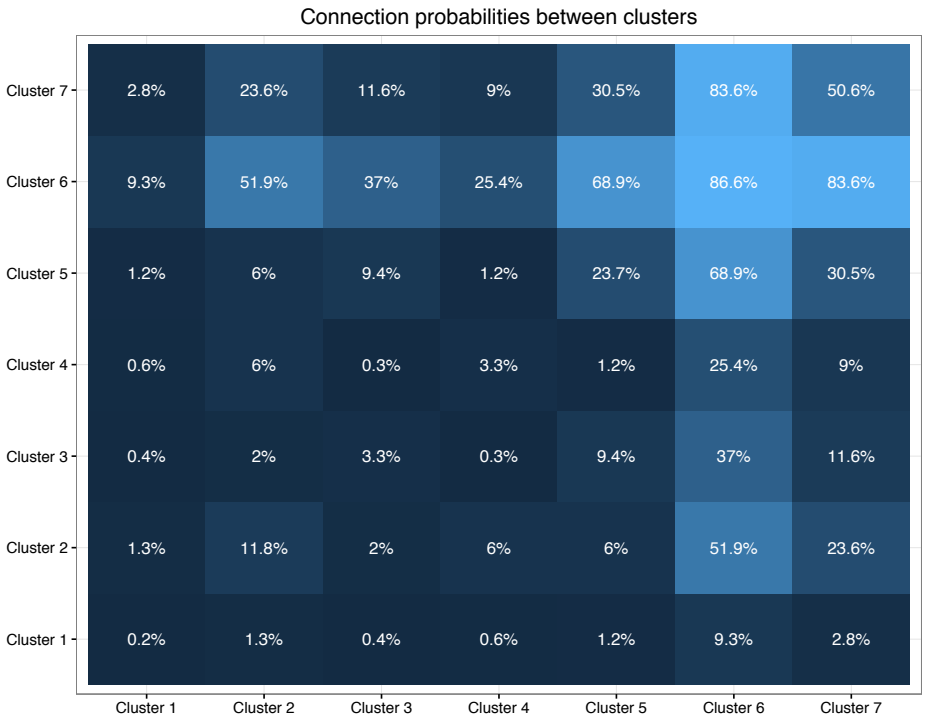


Figure 5.4: Terms Π^1_{kl} of the tensor matrix Π estimated using the VEM algorithm.

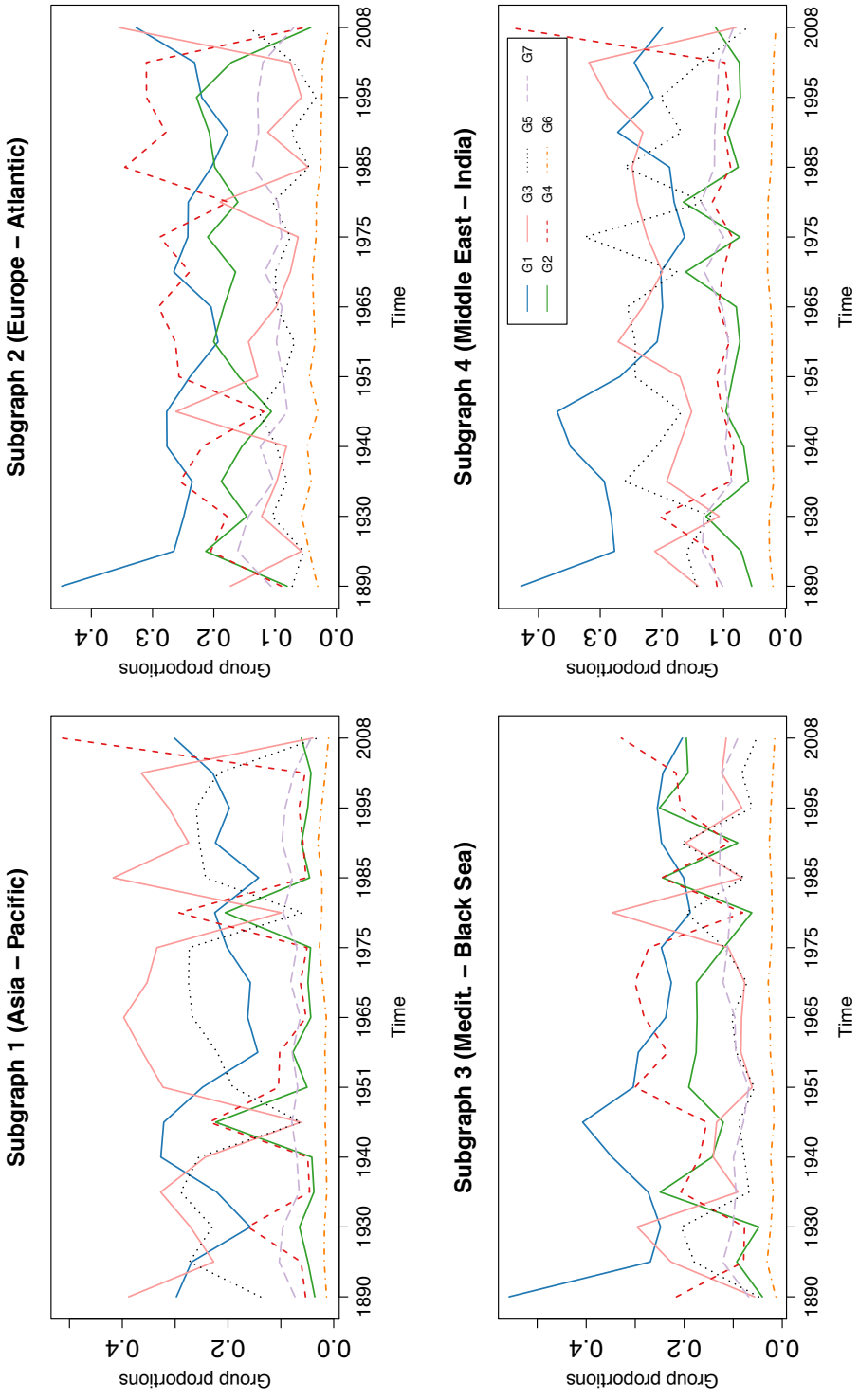


Figure 5.5: Evolution of the proportions of the $K = 7$ latent clusters.

Regarding cluster 7, one can see on Figure 5.5 that its proportions in the subgraphs are higher than those of cluster 6. The ports of cluster 7 can be qualified as hubs of second class which are subordinated to the main hubs of cluster 6. Most of them are marked by a colonial logic, such as Marseille, Kolkata or Cape Town. The evolution of this cluster until the recent period shows a persisting link North-South (*e.g.* Le Havre - Casablanca) or East-West (*e.g.* Spain - Brazil - Canaries).

Cluster 5 is mainly made of ports from the Asia – Pacific and Middle East – India basins except during major crises, such as World War II and the oil crisis. During those crises, the cluster mainly contains European ports. The rapid modification of this cluster appears clearly on Figure 5.5 around 1946, 1980 and 2008. This cluster can be interpreted as made of active ports from the developing world which move to cluster 2 during the crises. This may highlight the disintegration of long distance links during such crises. Conversely, cluster 2 turns out to be mostly made, except during crises, of European ports of average size, mainly on the atlantic coast. Those ports are rather a reflection of a past glory and most of them have declined over the century. This may due to a failed industrialization or a significant distance to the major trade routes.

Finally, clusters 3 and 4 are made of very small ports with low activity. Those ports are usually not connected together and communicate with the rest of the network only through ports of clusters 2 and 5. The connection with clusters 2 and 5 explains the brutal changes in the proportions of clusters 3 and 4 that one can also observe.

Partial maritime network

In this application, we focus on the collapse of the Union of Soviet Socialist Republics (USSR). Here, we shortcut the initial maritime data presented in the previous applications and we concentrate on the period around the occurrence of this event in 1991, studying it over 8 consecutive years (1987-1994). As this event impacted Socialist countries more then other, we decided to change the number of subgraphs from 4 to 3, to base them on 3 political categories: Capitalist, Socialist and Non-aligned. See Figure 5.7 showing the partition of ports into these 3 subgraphs on the map.

The USSR regime gradually influenced other socialist and communist countries over time. Figure 5.6 shows the spread of this influence to a number of countries. In the 20th century, the world's first constitutional socialist state was declared in Russia in 1917. Then, other former territories of the Russian empire joined it in 1922, thus becoming the USSR. In the aftermath of the second world war, the Soviet Army occupied much of Eastern Europe, thus helping to establish Communist states in these countries. Most of the Eastern Europe was allied to the USSR, except Yugoslavia which remained non-aligned. In 1949, the communist victory in China led to the establishment of the People's Republic of China. Cuba, Vietnam, Laos, and Cambodia followed later. Initially, North Korea also joined the communist states but then withdrew. In 1989, the Berlin's wall came down and with it the collapse of the Eastern European regimes as

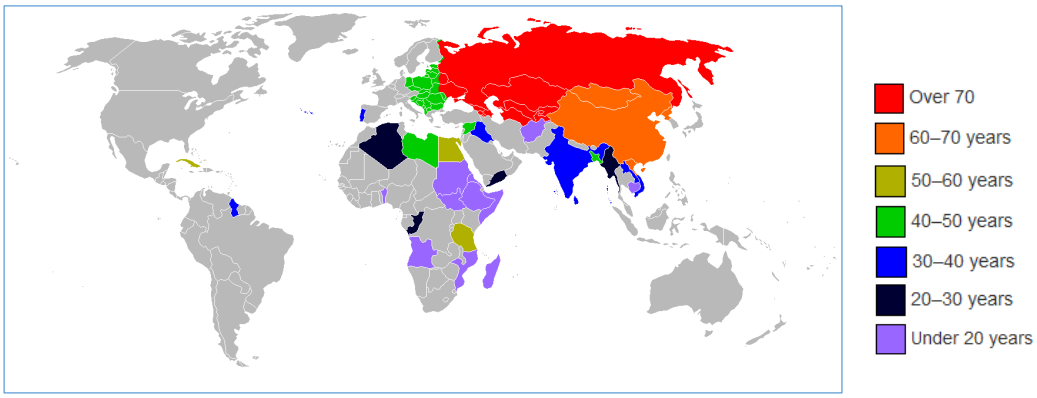


Figure 5.6: Countries that declared themselves socialist states under any definition, at some point in their history.

well as the break up of the Soviet Union in 1991. Today, the only remaining Communist states are China, Cuba, Laos, and Vietnam.

Let us note that in the previous study of the global maritime flows, this problem does not appear because the total number of ports was reduced and only the ports active in the 17 period were examined. For this reason, we have proposed the idea to applied our methodology in the same data with short duration.

Data and study protocol

At each time point, the adjacency matrix between ports is constructed, as in the previous study, for the global network where the entry in the adjacency matrix equals 1 if the number of ship movements between the two ports is greater or equal to 1, or equals 0 otherwise. Since, in this new data we have shortcut the time factor from 17 to 8 time period, the network size has grown in this application to include 2016 ports (out of 4472 ports worldwide) active throughout the whole period of this study.

Therefore, we have applied dRSM to a part of maritime network which describes the navigation of ships among 2006 ports in the world at 8 time points. Additionally, the partition of the network into subgraphs is provided here by the port memberships in the three main subgraphs namely: Capitalist, Socialist and Non-aligned. Figure 5.7 presents the partition of 2016 ports into subgraphs, each of a different colour. Hence, the distance between ports plays an important role, ship's weight can also be a major factor characterizing a group, in this context. We used this factor to show in Figure 5.9 the traffic percentages between subgraphs based on the ship's weight. On the basis of this graph, we can see the evolution and the dynamic of traffic between subgraphs as well as a shock,

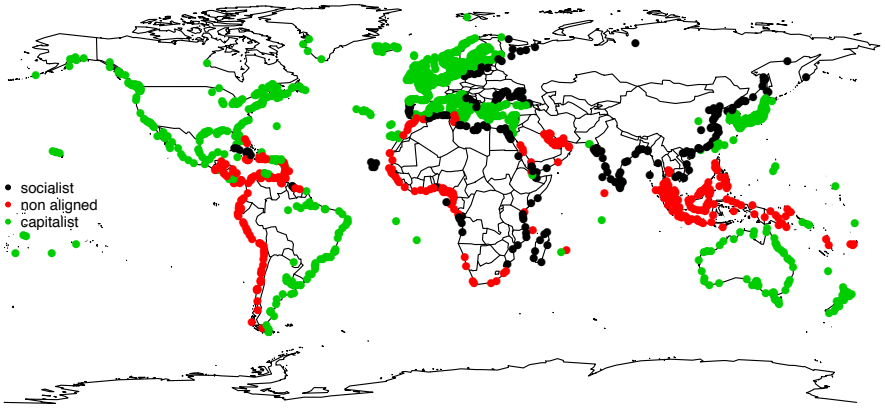


Figure 5.7: The given partition of the 2016 nodes (ports) into 3 subgraphs.

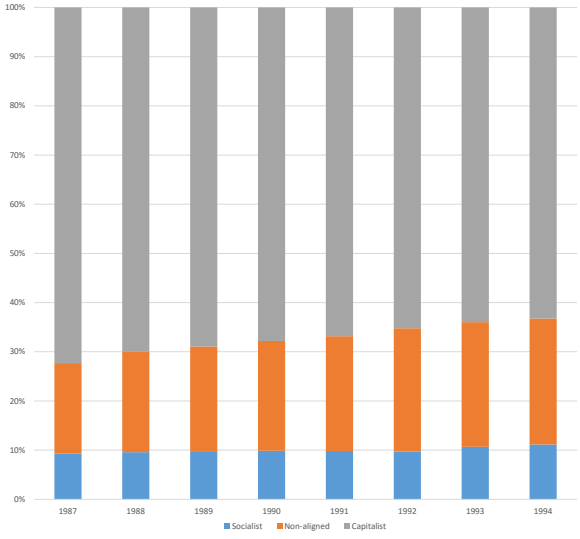


Figure 5.8: World maritime traffic.

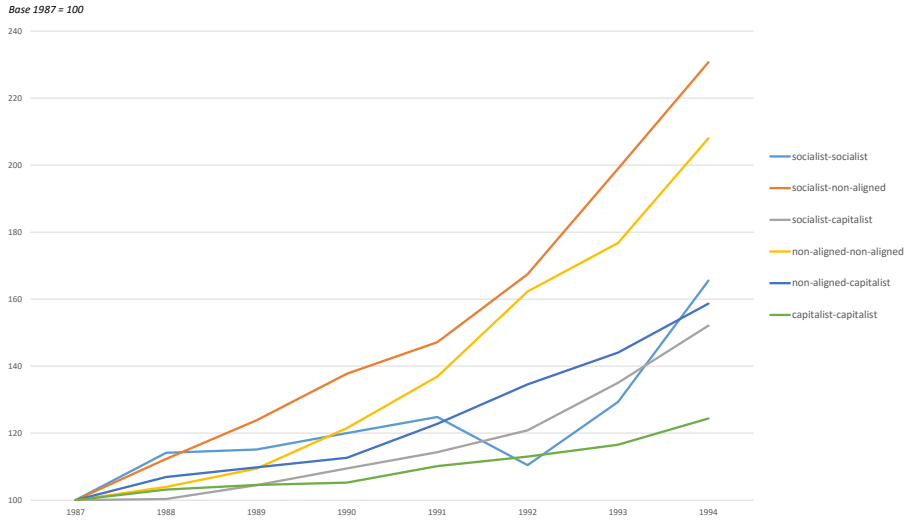


Figure 5.9: Inter-subgraph maritime flows.

specifically on the socialist-socialist traffic, due to the break up of the USSR which had a greater impact on the socialist countries. Additionally, Figure 5.8 shows the proportion of ports in each subgraph. Here we can see that the majority of the ports are in the Capitalist subgraph (1371 ports) versus 252 ports for Socialist ones. To summarize, our network is an undirected and binary network without self loops, i.e. C = 1 and X_{ij}^t=1 if the port i and the port j exchange at least one ship during the period t, 0 otherwise, with $t \in (1, \ldots, 8)$ and S = 3.

Results

To present the results of this application, we first show the selection of the number of groups. So, the choice of K is also made by applying the VEM algorithm for $K = 3, ..., 10$ and then by computing the associated BIC values. Once again, the retained value for K is the one associated with the highest BIC value. Still, to ensure a good accuracy of the results, the VEM algorithm was run 5 times for each value of K. Figure 5.10 shows the evolution of the BIC criterion according to K. One can observe that BIC peaks this time at $K = 9$, meaning that 9 latent groups seem to make up the network.

Second, in order to detect the characteristics of these 9 groups, we look for the estimated tensor matrix Π_{ql}^1 which describes the probabilities of connections between groups. This allows us to figure out if there are one or more dominant groups which have a strong connectivity to other groups. Similarly, we can search out groups with a low connectivity to others. Figure 5.11 shows the matrix Π_{ql}^1 to the different connection patterns between 9 groups for the binary

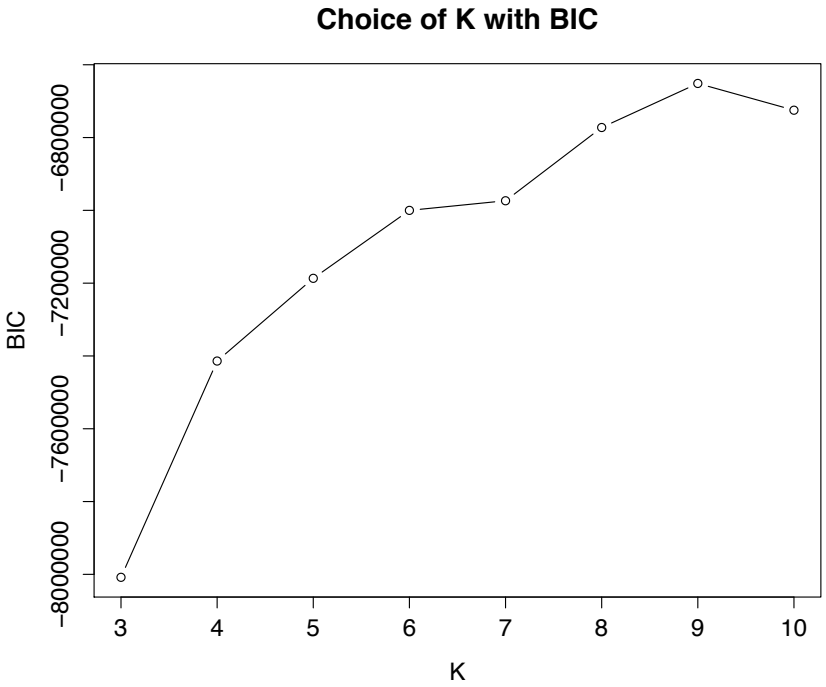


Figure 5.10: BIC values according to the number K of groups for the maritime network. The actual value for K is 9.

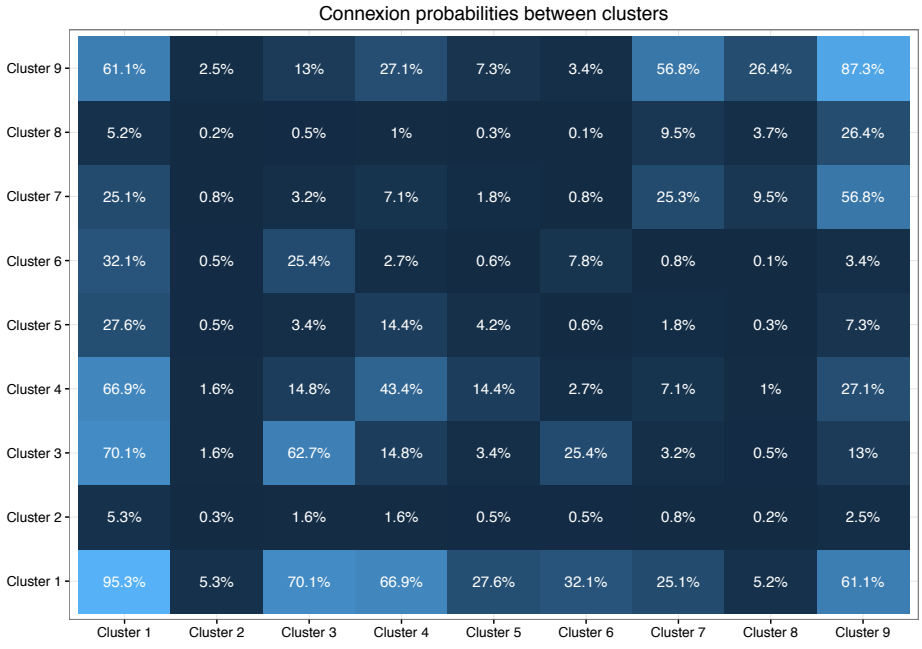


Figure 5.11: Terms π^1_{kl} of the tensor matrix π estimated using the VEM algorithm.

network. Therefore, regarding the result of this matrix, we note that we have two dominant clusters, 1 and 9, which have strong connections with all other clusters. Figure 5.14 presents again Π^1_{ql} in a different form to improve the observation of group connectivity.

In addition, to this matrix, we give the estimated group proportions for each subgraph over time, which are presented in Figure 5.13, to show the dynamic of this network. We see that in 1991, the year the USSR disintegrated, the graph shows some drops and peaks in certain clusters, while in other years, there occur distortions in proportions. For example, in clusters 4 and 5, there is a drop in Socialist and Non aligned subgraphs as against a peak in 1991 in clusters 3 and 6 of the same subgraphs. Also, we can see the opposite behavior in these groups in the Capitalist subraph. One also observes that the proportion of clusters 1 and 9 are low and that these are groups with a limited number of hubs and high connectivity with all other groups. It leads us to conclude that they have a high level of traffic which is confirmed by Figure 5.11 and Figure 5.14. So, our dRSM model clearly detected that there is a problem of traffic in 1991 and distortions in the proportion of clusters around this year.

On the other hand, from the point of view of geography, the 9 clusters can be split into 3 different categories of ports: big, medium and small, more or less impacted by the shock. The existence of many groups in different categories is due to several reasons, such as port's geographical location and the type of ship

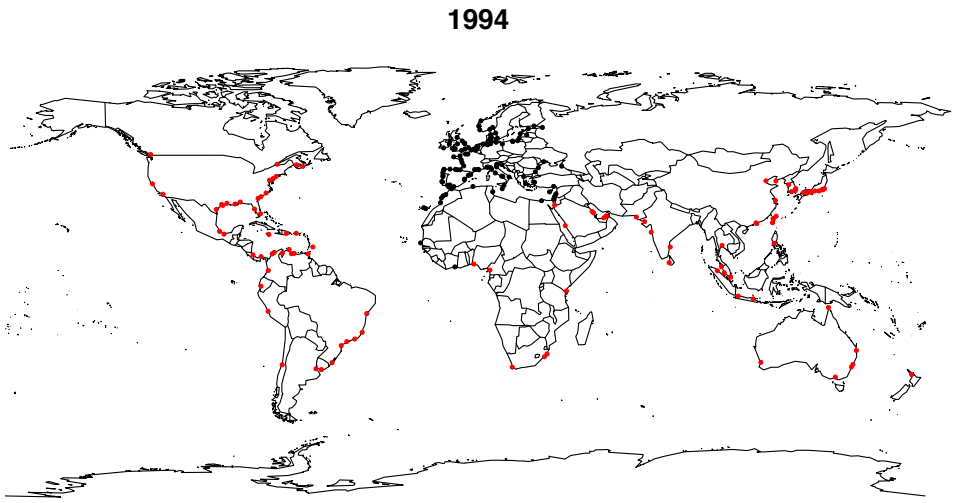


Figure 5.12: The proportions of the latent clusters 1 and 9.

traffic, i.e. directed link or indirect link (ports of call on the way). Therefore, clusters 1 and 9 contain big ports, less vulnerable to the shock. The difference between these two groups lies in the geographical locations and the directed or indirected link of the traffic between ports. These factor serves to separate the majority of big ports from each other (see Figure 5.12). Cluster 1 includes big ports in and around Western Europe where majority of the traffic is direct, such as Rotterdam, Barcelona, London, Bordeaux, etc. They remained part of the cluster even in 1991. Whereas, some other ports from cluster 1 moved to cluster 9 in 1991, Singapore for instance. Cluster 9, includes big ports all over the world with indirect traffic, such as Tokyo, Bangkok, Aukland, etc. Furthermore, on the basis of these results, it can be observed that small ports in cluster 2, 7, and 8 are also, less likely, to be impacted by the shock, when there is low connectivity to the main ports in cluster 1 and 9.

Finally, Clusters 3-6 and 4-5 containing medium sized ports were more influenced by the shock than the others. Figure 5.16 presents the configuration of these ports on the map where each colour corresponds to a single cluster. The shift in colors in 1991 is remarkable where there is a change in trade between ports in these groups. These clusters are politically specialized, see north Europe (clusters 3 and 6, which are traditionally more capitalistic) shifted to clusters 4 and 5 (receiving more socialist traffic) in 1990-1991. The opposite is true for the Atlantic region, where the ports are traditionally more socialist and non-aligned, they shifted to clusters 3 and 6 in 1990-1991. To present additional results, we show in Figure 5.17 the following ratio:

$$\frac{\text{Traffic percentage of each cluster in each subgraph}}{\text{global traffic percentage in each subgrapgh}}$$

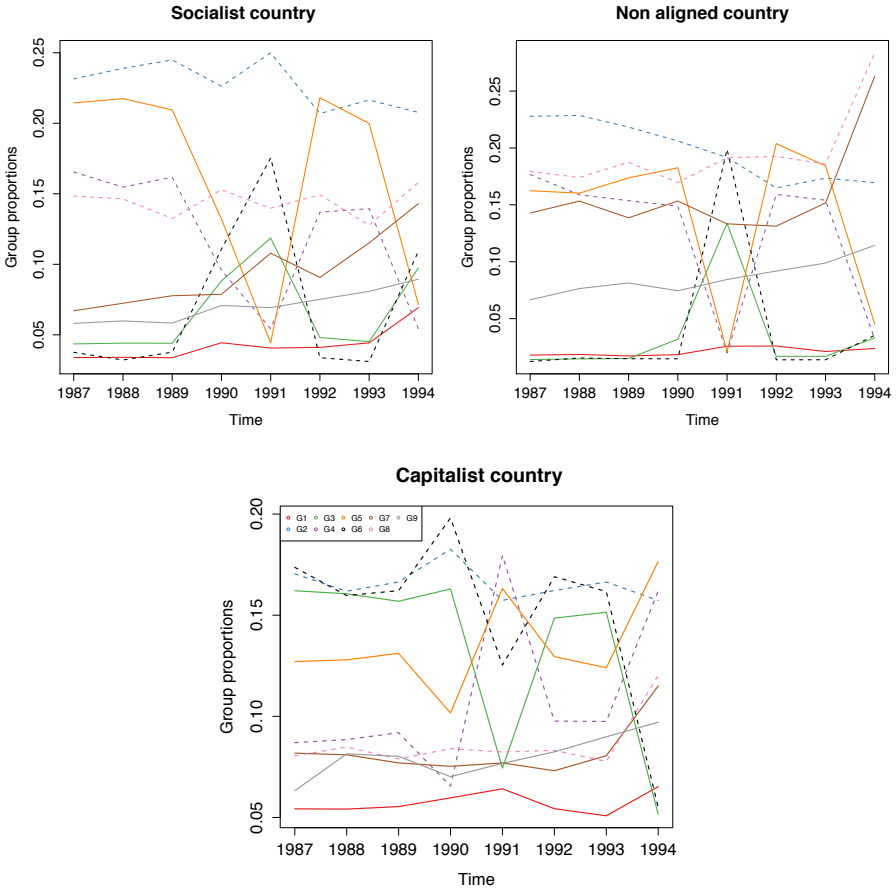


Figure 5.13: Evolution of the proportions of the $K = 9$ latent clusters .

This ratio based on the weights measured by the deadweight tonnage (dwt) allows us to confirm the specialization of clusters over time. This graph indicates that clusters 1-3-6 are specialized in the "capitalist" traffic, and the clusters 2-7-8 in the "non-aligned" traffic. Whereas clusters 4-5-9 receive more "socialist" traffic (with evolution nets). This ratio confirms our classification for each cluster. Consequently, the USSR lost its longer-distance maritime linkages (including Cuba) in 1990-1991 and reshifted its influence over nearby Europe (Northwest).

Further research: this application leads to other applications of dRSM owing to the need to look at specific trades (e.g. bulks) and specific fleets (e.g. Soviet vs. capitalist), where we consider that the edges are categorical depending on ship's weight.

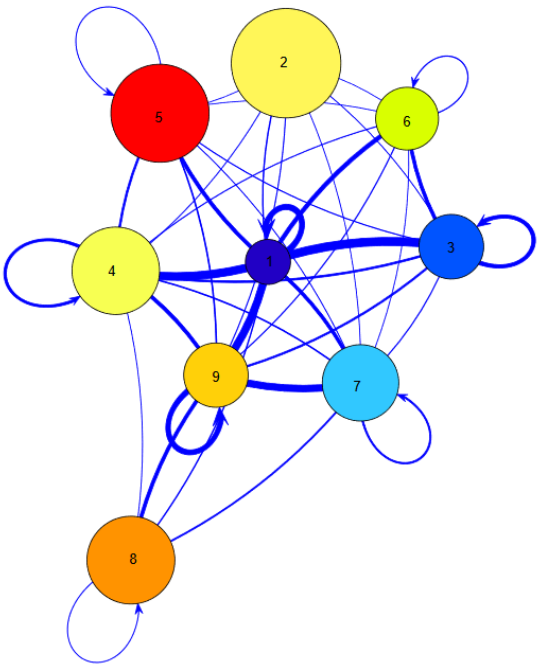


Figure 5.14: Summary of connection probabilities between groups.

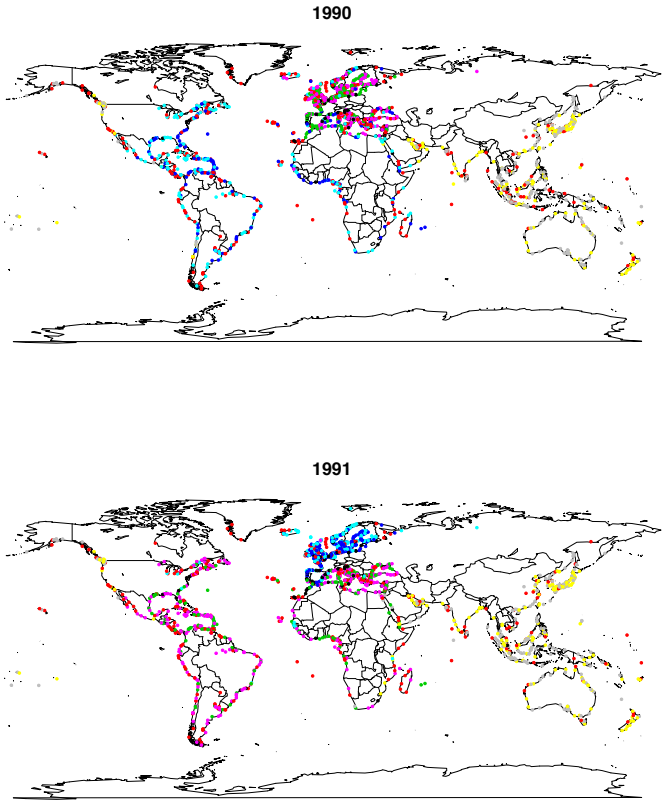


Figure 5.15: The partition of ports into 9 groups (colors) during 2 different years (1990-1991).

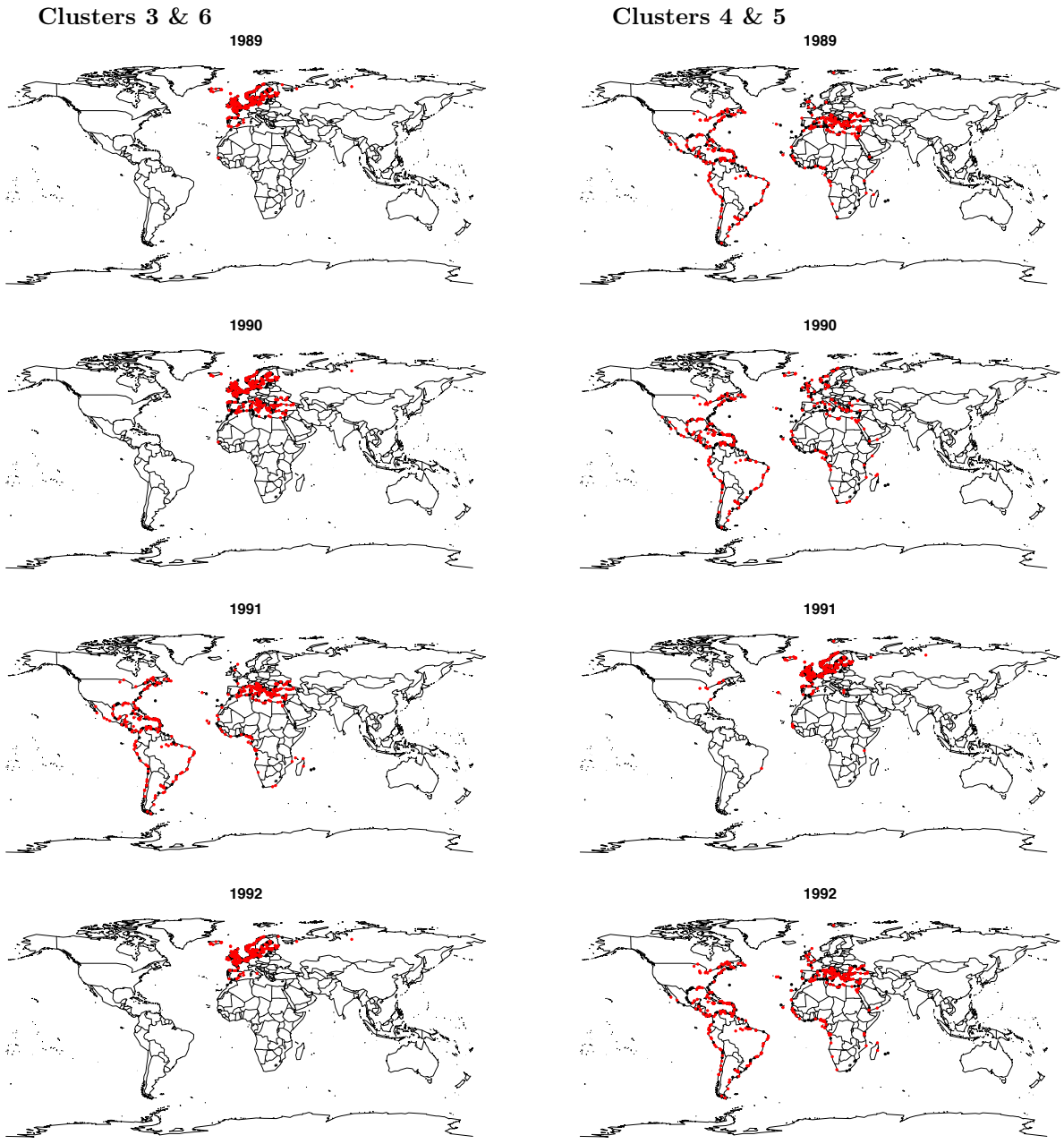


Figure 5.16: Cluster geography during the 4 years before and after USSR collapse for clusters 3,6 and 4,5.

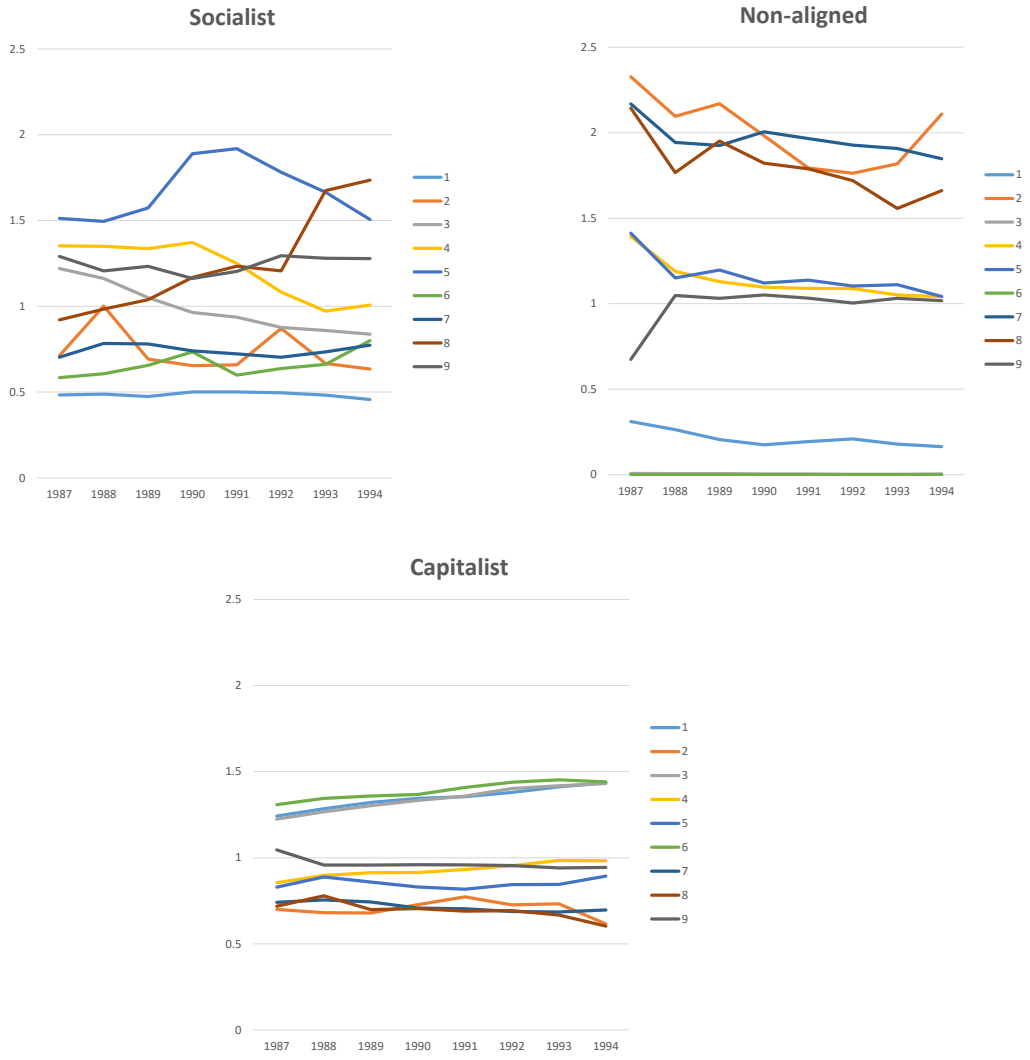


Figure 5.17: Cluster evolutions in fonction of its ratio of weights.

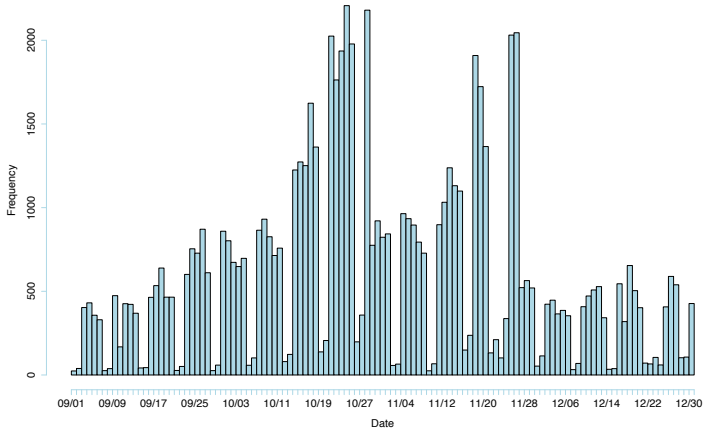


Figure 5.18: Frequency of messages between Enron employees between September 1st and December 31th, 2001.

5.1.2 Application to the Enron network

Here we study a classical communication network, the Enron data set. This set contains all email communications between 149 employees of the famous company from 1999 to 2002. The original data set is available at `https://www.cs.cmu.edu/~./enron/`. The period under consideration is September, 1st to December, 31th, 2001. We focused on this specific time window because it is the densest period in terms emails sent, as it corresponds to a critical period for the company. Indeed, after the announcement early September 2001 that the company was in the strongest and best shape than it had ever been, the Securities and Exchange Commission (SEC) opened an investigation on October, 31st for fraud. The company finally filed for bankruptcy on December, 2nd, 2001. By this time, it became the largest bankruptcy in U.S. history and resulted in more than 4,000 job losses. Unsurprisingly, these key dates actually correspond to breaks in the email activity of the company, as shown by Figure 5.18.

Data study and protocol

The Enron data set, is a network which describes the exchange of emails among 148 individuals who worked for Enron company at each time t. In order to show the changes in structural developments within the company, data is first grouped by month, followed by periods of time, in a way that two persons are considered connected if they have exchanged at least one email during the period under consideration. We are interested here in five time periods noted $t_1, t_2, \ldots, t_5$ (Table 5.2) including the key events in the Enron scandal. We have divided the periods, in such a way so as to keep sufficient network density for each

t	périodes
t_1	du 01/01/2000 au 01/12/2000
t_2	du 01/01/2001 au 01/03/2001
t_3	du 01/04/2001 au 01/06/2001
t_4	du 01/07/2001 au 01/09/2001
t_5	du 01/10/2001 au 01/03/2002

Table 5.2: The time periods considering for the analysis of e-mail exchanges in the Enron company

time period. The transactions undertaken to hide the losses incurred through speculation were made in October 2001, in the aftermath of an investigation carried out by the energy regulation agency. The company, finally, went bankrupt in 2001. These two events correspond to the period t_5.

On the other hand, we propose a partition of the employees into three subgraphs depending on their status in the company (s_1 : Managers, s_2 : Employees, s_3 : Others). It should be mentioned that sungraph s_1 contains all the managers of the company, that is to say, the general director, presidents, vice presidents, directors, managers and directors of managers. The traders are also incorporated in this subgraph. The individuals of subgraph s_2 are all the employees of company excepting the managers. Finally, s_3 represents all other individuals in contact with Enron company who were also affected during the crisis of October 2001.

Briefly, the network is a directed and binary network without self loops, *i.e.* $C = 1$ and $X_{ij}^t = 1$ if i and j exchanged at least one email during the period t, 0 otherwise, with $t \in \{1, \ldots, 5\}$ and $S = 3$. We consider also the known partition of network into three sub-graphs corresponding to the status of the employees in the company.

Results

The VEM algorithm which was introduced in Section 2.6.1 and chosen for approximation of our dRSM model has been applied to the data set in order to look for $K = 4$ latent groups. This choice was dictated by empirical considerations and allows to obtain a dRSM model describing the emergence and especially, the crisis management following the start of the investigation.

Among all the results obtained through our approach, firstly, we take an interest in the topology of the latent groups found. The network is binary, the dRSM model used here is a special case of the model we proposed with $C = 1$. By construction, the matrix Π verifies $\Pi_{kl0} + \Pi_{kl1} = 1$, $\forall\,(k, l)$ therefore, only the Π_{kl1} terms describing the connection probabilities are given in Table 5.3. Table 5.3 shows that the three clusters (1, 2 and 4) correspond to the communities where the probability of connections between two nodes of the same community is stronger than between different communities' nodes. Thus, these clusters are mainly distinguished by the fact that they have different intra-cluster

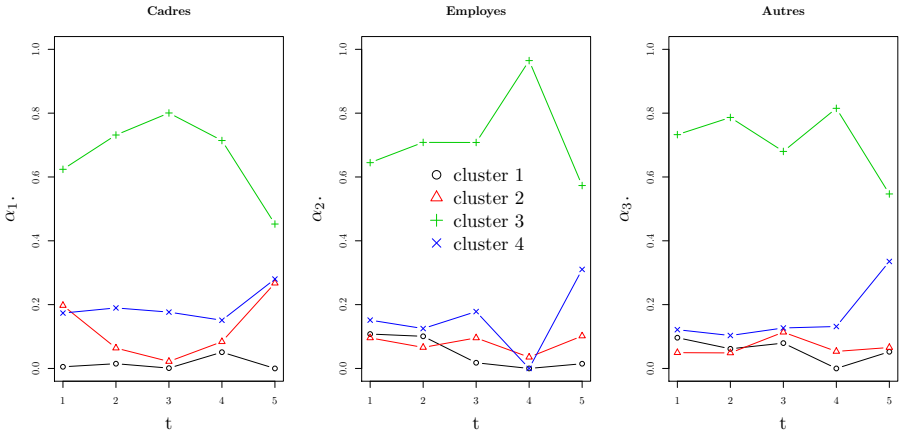


Figure 5.19: Proportions of the $K = 4$ clusters, at each time. Subgraph 1 (Managers), left figure; subgraph 2 (employees), middle figure; subraph 3 (other), right figure.

	cluster 1	cluster 2	cluster 3	cluster 4
cluster 1	0.478	0.037	0.005	0.023
cluster 2	0.020	0.181	0.006	0.012
cluster 3	0.001	0.002	0.001	0.003
cluster 4	0.012	0.012	0.024	0.119

Table 5.3: Terms Π_{kl1} of the matrix Π estimated using the VEM algorithm

probabilities of connection, where cluster 1 has the highest density (Π^1_{11}=0.478), followed by clusters 2 and 4 . Finally, cluster 3 is built from low probabilities of connection (0.001). It gathers in fact at all individuals participating in non structured exchange in the network.

Applying our dRSM model on the Enron network allows us to characterize the evolution of the subgraphs with latent clusters according to time. Figure 5.19 presents all estimated proportions of three subgraphs. Like the previous results, this Figure shows that cluster 3 gather the nodes loosely structured in the network. The individuals associated with this cluster at the time t exchange emails with other individuals in the network, without connection profile type. See that, increasing proportions of this cluster are coincide with the decrease in the proportions of cluster 2 (average intra-cluster density), regardless of the subgraphs or time t, and vice versa. Therefore, these two proportions inform the inverse display on the structure of emails exchanged in the network.

We observe a drop in the proportion of cluster 3, in all subgraphs between t_4 and t_5, *i.e.* just before and after the opening of the investigation by the US federal agency. This specific network structure here is a reaction to the crisis of October 2001. The employees exchanged emails on the subject and

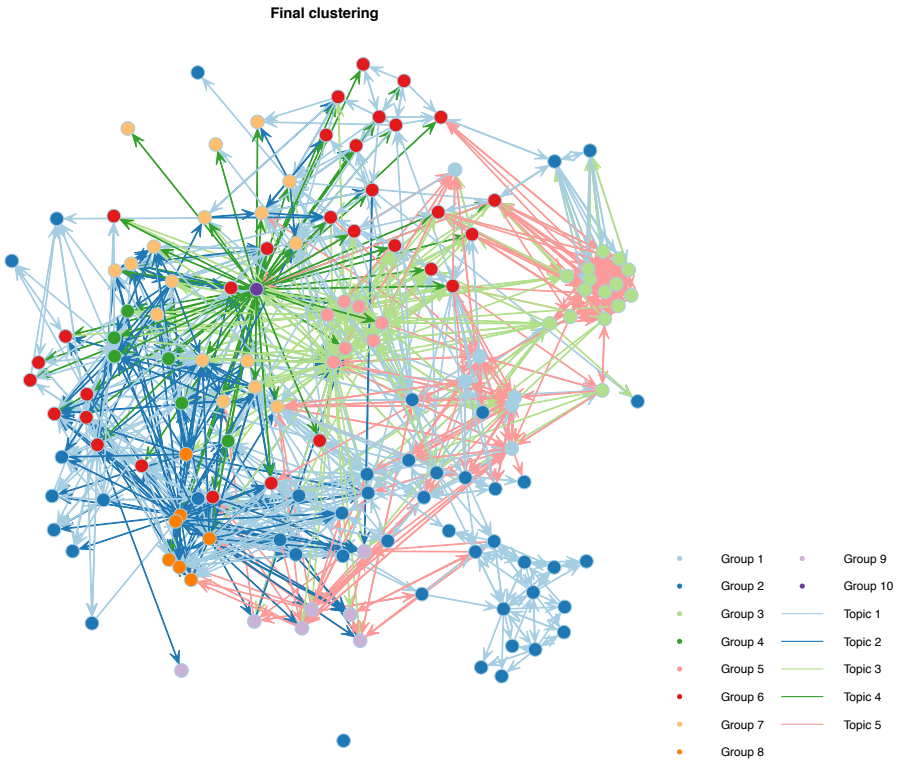


Figure 5.20: Clustering result with STBM on the Enron data set (Sept.-Dec. 2001).

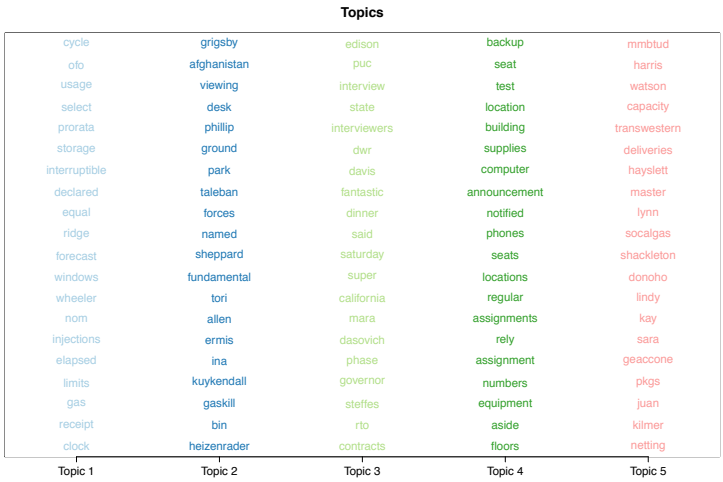


Figure 5.21: Most specific words for the 5 found topics with STBM on the Enron data set.

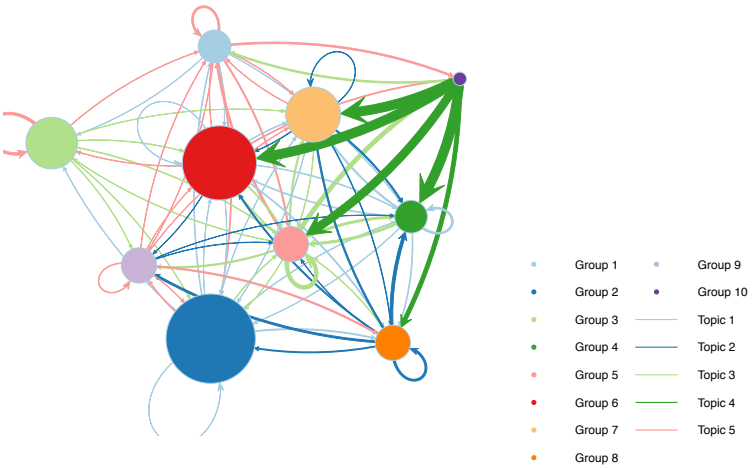


Figure 5.22: Enron data set: summary of connexion probabilities between groups (π, edge widths), group proportions (ρ, node sizes) and most probable topics for group interactions (edge colors).

contacted people preferentially. At this period, the proportion of cluster 4 (lower intra-cluster density), such as the one in cluster 3, increase. It is worth noticing that the structuring of the network starts earlier (at t_3) among managers than among employees. The subgraphs 2 and 3 have a mild reaction to others, but it disappears at t_4. This observation suggests that managers were aware of the arrival of the crisis before other employees. Now we concentrate on cluster 1 with high intra-cluster density, which allows us to see that the managers are the only individuals in the network for which we have observed a fall in the proportion of cluster 1 at the time (t_4, t_5) unlike the subgraphs 2 and 3. That remark also goes in the direction of a network structure related to the opening of the investigation. The cadres are the only individuals in the network for whom we observe the contrary at the time of a decrease in the proportion of Group 1. Through these observations and by taking into consideration the high position of managers where the exchange of emails is carried out on a very preferential basis. This allows us to separate the managers from the rest of the network. Finally, the estimated matrix Φ has the form: $\Phi = \sigma^2 I_{K-1}$ with $\sigma^2 = 7.11$. Additionally, this data is, therefore, characterized by a strong variance of the time process.

5.2 STBM applications

In this section, we present two applications of STBM, introduced in Chapter 5 to real-world networks: the Enron email and the Nips co-authorship networks. These two data sets have been chosen because one is a directed network of moderate size whereas the other one is undirected and of a large size.

5.2.1 Enron email network analysis

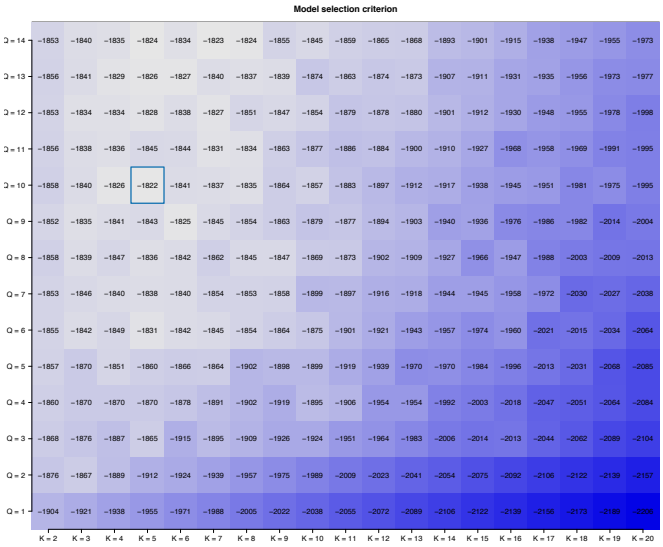


Figure 5.27: Model selection for STBM on the Enron data set.

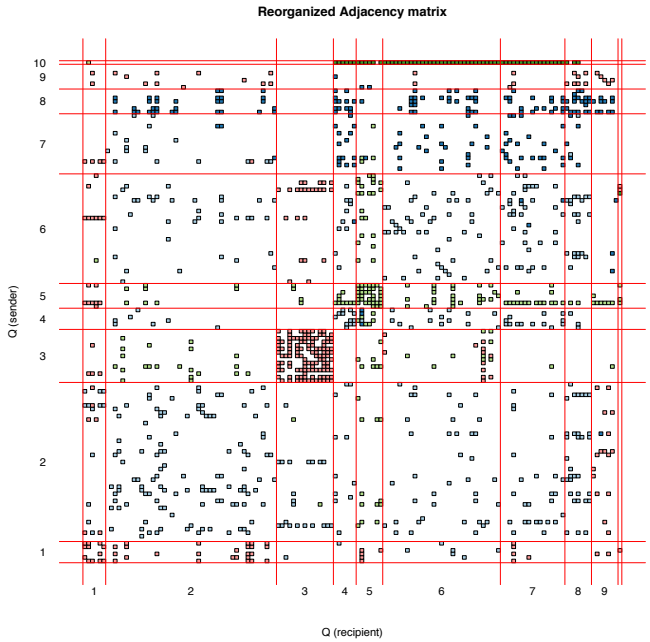


Figure 5.28: Reorganized adjacency matrix according to groups for STBM on the Enron data set.

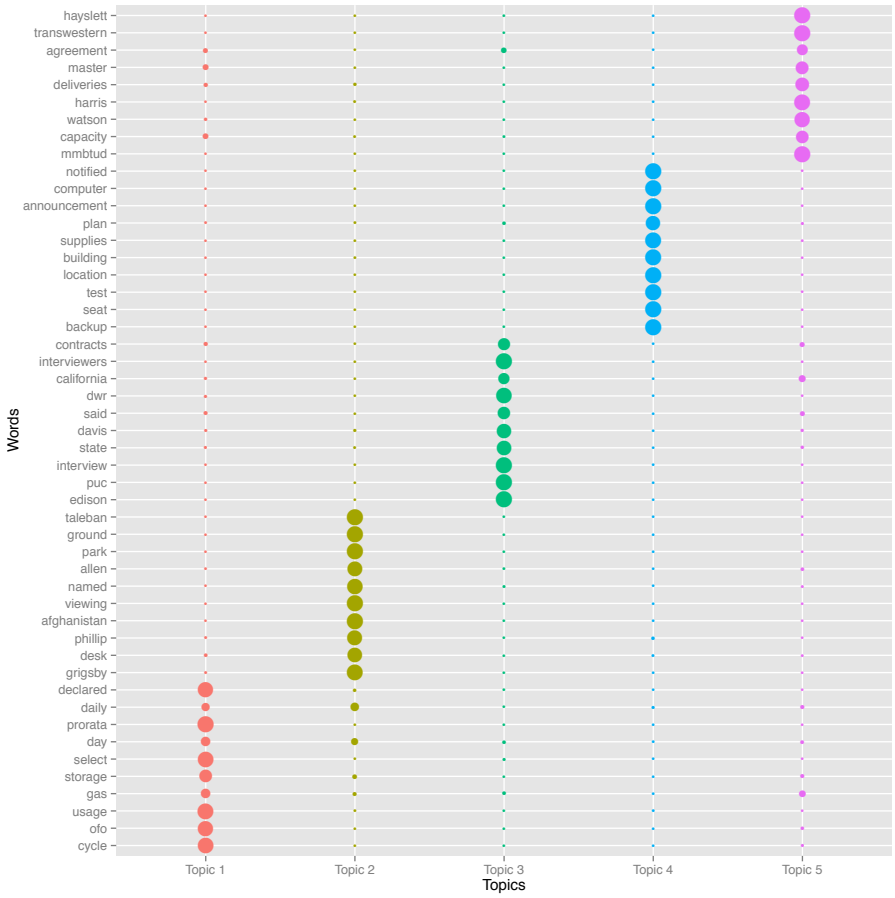


Figure 5.29: Specificity of a selection of words regarding the 5 found topics by STBM on the Enron data set.

Here, we applied STBM on the Enron data set which was used and described in Section 5.1.2. As a reminder, the data set contains all email communications between 149 employees. The data set considered here contains 20 940 emails sent between the $M = 149$ employees. All messages sent between two individuals were coerced in a single meta-message. Thus, we end up with a data set of 1 234 directed edges between employees, each edge carrying the text of all messages between two persons.

The C-VEM algorithm we developed for STBM was run on these data for a number Q of groups from 1 to 14 and a number K of topics from 2 to 20. As one can see on Figure 5.27, the model with the highest value was $(Q, K) = (10, 5)$. Figure 5.20 shows the clustering obtained with STBM for 10 groups of nodes and 5 topics. As previously, edge colors refer to the majority topics for the communications between the individuals. The found topics can be

easily interpreted by looking at the most specific words of each topic, displayed in Figure 5.21. In a few words, we can summarize the found topics as follows:

- Topic 1 seems to refer to the financial and trading activities of Enron,
- Topic 2 is concerned with Enron activities in Afghanistan (Enron and the Bush administration were suspected to work secretly with Talibans up to a few weeks before the 9/11 attacks),
- Topic 3 contains elements related to the California electricity crisis, in which Enron was involved, and which almost caused the bankruptcy of SCE-corp (Southern California Edison Corporation) early 2001,
- Topic 4 is about usual logistic issues (building equipment, computers, ...),
- Topic 5 refers to technical discussions on gas deliveries (mmBTU represents 1 million of British thermal unit, which is equal to 1055 joules).

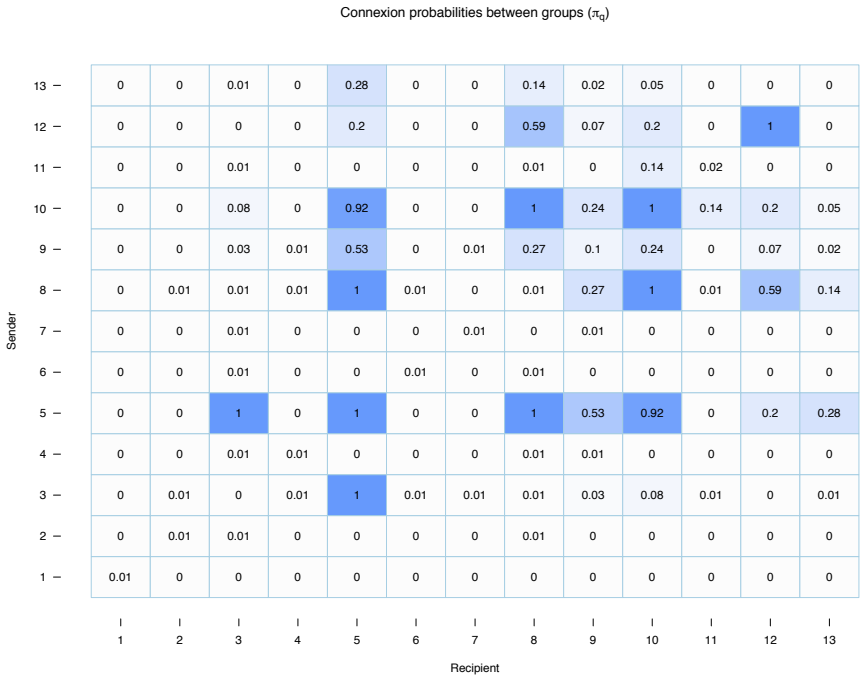


Figure 5.30: Estimated matrix π by STBM on the Nips co-authorship network.

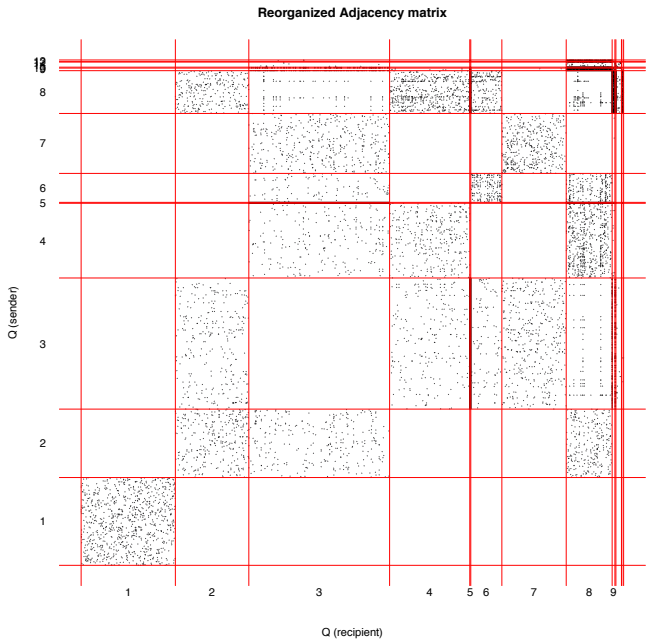


Figure 5.31: Reorganized adjacency matrix according to groups for STBM on the Nips co-authorship network.

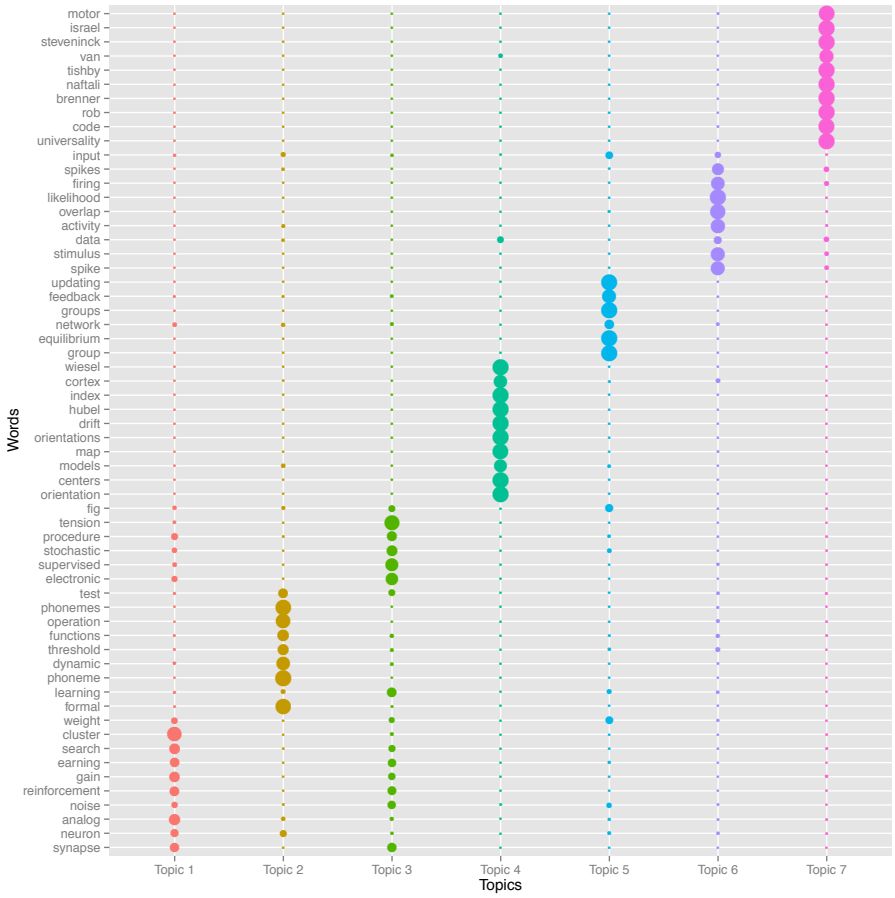


Figure 5.32: Specificity of a selection of words regarding the 5 found topics by STBM on the Nips co-authorship network.

Figure 5.22 presents a visual summary of connexion probabilities between groups (the estimated π matrix) and majority topics for group interactions. A few elements deserve to be highlighted in view of this summary. First, group 10 contains a single individual who has a central place in the network and who mostly discusses about logistic issues (topic 4) with groups 4, 5, 6 and 7. Second, group 8 is made of 6 individuals who mainly communicates about Enron activities in Afghanistan (topic 2) between them and with other groups. Finally, groups 4 and 6 seem to be more focused on trading activities (topic 1) whereas groups 1, 3 and 9 are dealing with technical issues on gas deliveries (topic 5).

As a comparison, the network has also been processed with SBM, using the mixer package (Ambroise et al., 2010). The chosen number K of groups by SBM was 8. Figure 5.23 allows to compare the partitions of nodes provided by SBM and STBM. One can observe that the two partitions differ on several points.

On the one hand, some communities found by SBM (the bottom-left one for instance) have been split by STBM since some nodes use different topics than the rest of the community. On the other hand, SBM isolates two "hubs" which seem to have similar behaviors. Conversely, STBM identifies a unique "hub" and the second node is gathered with other nodes, using similar discussion topics. STBM has therefore allowed a better and deeper understanding of the Enron network through the combination of text contents with network structure.

5.2.2 Nips co-authorship network analysis

This second network is a co-authorship network within a scientific conference: the Neural Information Processing Systems (Nips) conference. The conference was initially mainly focused on computational neurosciences and is nowadays one of the famous conferences in statistical learning and artificial intelligence. We here consider the data between the 1988 and 2003 editions (Nips 1–17). The data set, available at `http://robotics.stanford.edu/~gal/data.html`, contains the abstracts of 2 484 accepted papers from 2 740 contributing authors. The vocabulary used in the paper abstracts has 14 036 words. Once the co-authorship network reconstructed, we have an undirected network between 2 740 authors with 22 640 textual edges.

We applied STBM on this large data set and the selected model by ICL was $(Q, K) = (13, 7)$. Figure 5.24 shows the clustering obtained with STBM for 13 groups of nodes and 7 topics. Due to size and density of the network, the visualization and interpretation from this figure are actually tricky. Fortunately, the meta-view of the network shown by Figure 5.25 is of a greater help and allows to get a clear idea of the network organization. To this end, it is necessary to first to picture out the meaning of the found topics (see Figure 5.26):

- Topic 1 seems to be focused on neural network theory, which was and still is a central topic in Nips,
- Topic 2 is concerned with phoneme classification or recognition,
- Topic 3 is a more general topic about statistical learning and artificial intelligence,
- Topic 4 is about Neuroscience and focuses on experimental works about the visual cortex,
- Topic 5 deals with network learning theory,
- Topic 6 is also about Neuroscience but seems to be more focused on EEG,
- Topic 7 is finally devoted to neural coding, *i.e.* characterizing the relationship between the stimulus and the individual responses.

In light of these interpretations, we can eventually comment some specific relationships between groups. First of all, we have an obvious community (group 1) which is disconnected with the rest of the network and which is focused on

neural coding (topic 7). One can also clearly identifies, on both Figure 5.25 and the reorganized adjacency matrix (Figure 5.31), that groups 2, 5 and 10 are three "hubs" of a few individuals. Group 2 seems to mainly work on the visual cortex understanding whereas group 10 is focused on phoneme analysis. Group 5 is mainly concerned with the general neural network theory but has also collaborations in phoneme analysis. From a more general point of view, topics 6 and 7 seem to be popular themes in the network. It is also of interest to notice that statistical learning and artificial intelligence (which are probably now 90% of the submissions at Nips) were not yet by this time proper thematics. They were probably used more as tools in phoneme recognition studies and EEG analyses. This is confirmed by the fact that words used in topic 3 are less specific to the topic and are frequently used in other topics as well (see Figure 5.32).

As a conclusive remark on this network, STBM has proved its ability to bring out concise and relevant analyses on the structure of a large and dense network. In this view, the meta-network of Figure 5.25 is a great help since it summarizes several model parameters of STBM.

5.3 Conclusion

This chapter has presented « the fruits of our labor », where we have applied our two new models (dRSM and STBM) to several real data sets. These applications illustrate the capacities and the usefulness of each of our models. First, the dRSM model turned out to be able to efficiently recover the dynamic processes of a communication network and a geographical network. In particular, dRSM spotted interesting events like the breakup of the Soviet Union and its impact on the maritime flows. These results show the great potential of this kind of statistical tools for geographers, and more widely for researchers in digital humanities. Second, the applications of STBM presented in this chapter proved its ability to deal with a type of data which were not considered until now in the statistical framework. The analysis with STBM of an e-mail network and a co-authorship network allowed to recover clusters of individuals which are coherent from both the network activity and the text content. Indeed, STBM allows the two data types to enrich each other and to provide more accurate results. This kind of solutions may be particularly valuable for social network analysts.

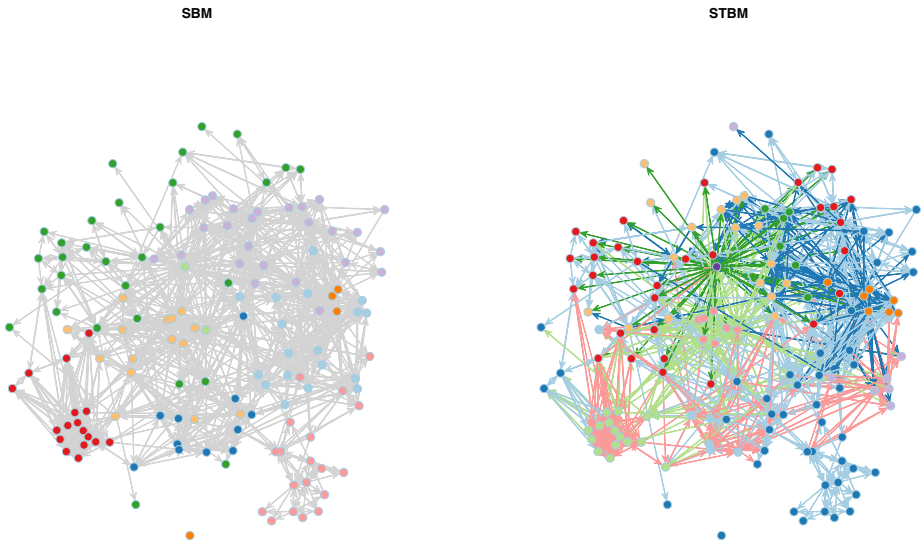


Figure 5.23: Clustering results with SBM (left) and STBM (right) on the Enron data set. The selected number of groups for SBM is $Q = 8$ whereas STBM selects 10 groups and 5 topics.

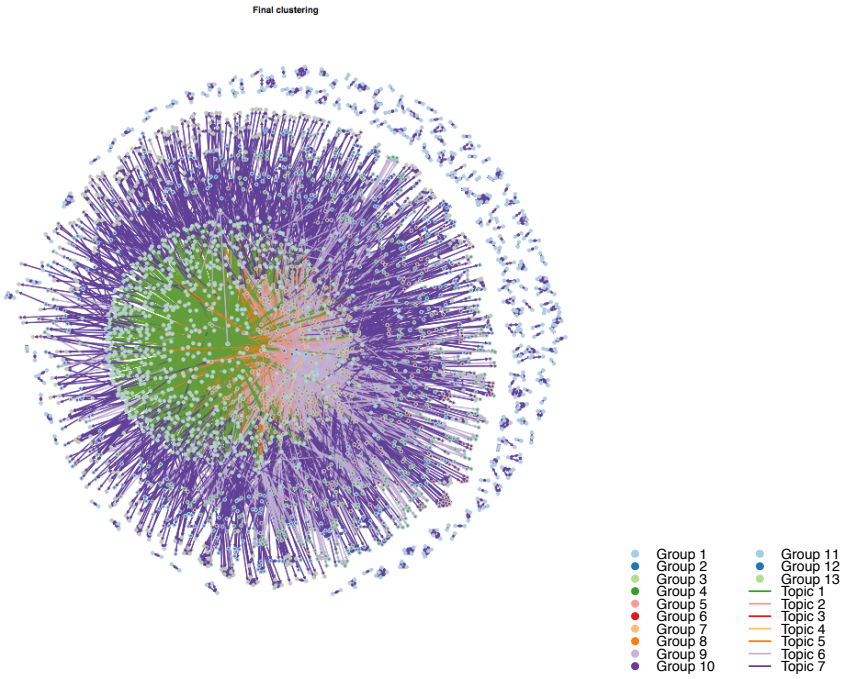


Figure 5.24: Clustering result with STBM on the Nips co-authorship network.

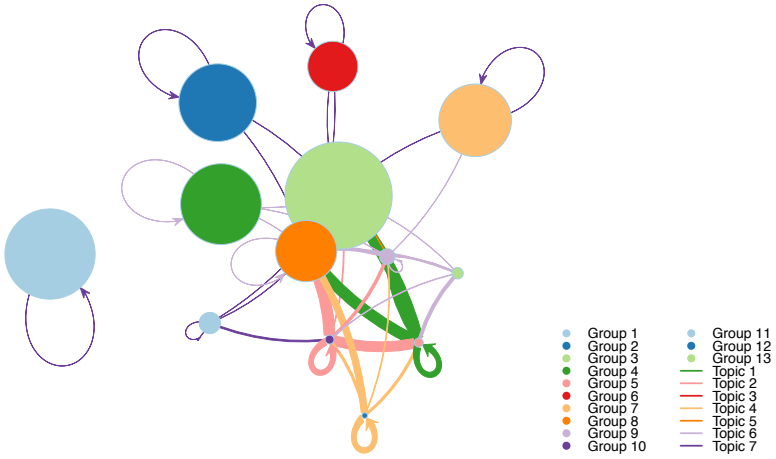


Figure 5.25: Nips co-authorship network: summary of connexion probabilities between groups (π, edge widths), group proportions (ρ, node sizes) and most probable topics for group interactions (edge colors).

Figure 5.26: Most specific words for the 5 found topics with STBM on the Nips co-authorship network.

CHAPTER 6

CONCLUSION AND PERSPECTIVES

The need to understand and analyze the structure of networks has been gaining in importance in various fields of research, especially where there is interaction between nodes in different forms. The study of network structures has been focusing on extracting as much hidden information as possible. In this thesis, we have proposed two new statistical models which provide answers to several queries regarding network analysis: the modeling and the clustering of dynamic networks and networks with textual edges.

6.1 Contributions of the thesis

The first contribution of this thesis extends the RSM model to the dynamic framework dRSM method which extends the RSM model to the dynamic framework. Thus, dRSM is able to model and cluster dynamic networks with categorical edges where an external partition of the network into subgraphs is known. We proposed a state space model to control the evolution of the latent group proportions over time. Model inference is done using variational expectation-maximization (VEM) algorithm and the BIC criterion is used for selecting the number of clusters.

Our second contribution was to consider a type of networks, networks with textual edges, which was never examined in the statistical literature and to propose the STBM methodology to deal with it. STBM mixes the principles of SBM, for the network part, and LDA for the text part. Regarding model inference, we introduced an original algorithm, the classification variational expectation-maximization (CVEM) algorithm. The selection of both the numbers of groups and topics relies on the ICL criterion. Thus, STBM allows to find clusters of individuals while identifying the discussion topics. The use of STBM permits to

recover clusters of individuals which are coherent from both the network activity and the text content.

The last contribution of this work is to have shown the capacity of the proposed methodologies to be used on digital humanities problems. Indeed, we applied dRSM and STBM in maritime geography and social network analysis. On the one hand, dRSM appeared as a useful tool for geographers since it succeeded in identifying interesting events like the breakup of the Soviet Union and its impact on maritime flows. On the other hand, STBM has shown its interest when analyzing the Enron e-mail network by detecting groups of employees related with either financial or political plots.

6.2 Perspectives

Since statistical network analysis is a quite recent research field, many perspectives can be envisaged on the basis of this thesis. We detail hereafter some perspectives, from either the methodological or application points of view.

6.2.1 Methodological perspectives

Node evolution in dRSM The dRSM model is a new model which handles the evolution of connections between nodes over time. In its current form, dRSM assumes however that both the number of groups and the nodes are fixed. Unfortunately, in many practical cases, the appearance and the disappearance of nodes over time are possible. For instance, the original maritime flow network contains several ports which disappeared in the studied period. Consequently, it would be important to be able to model the node evolution. A natural way to handle this evolution would be introduce a Markov chain or a birth-death process (Greene et al., 2010). The choice of the approach would depend on the type of time dynamic: the birth-death process is design of the continuous time modeling whereas a Markov chain would be more appropriate for discrete times. In practice, the birth-death process would be useful for long time networks such as historical and geographical networks. Conversely, the Markov chain would be pertinent for short time networks, such as social networks where people alternates frequently between different status (work, vacation, maternity leave, sick leave, ...).

Group evolution in dRSM Similarly, the number of groups present in the networks may evolve over time. This would be particularly if there are significant changes in the number of nodes. For instance, when considering a historical network, it may be possible to see the disappearance of groups after a major event such as a war. In order to take into account the group evolution in the model, we may introduce once again a birth-death process (Kim and Leskovec, 2013b) or even a Dirichlet mixture process such as the infinite mixture model (Ghahramani and Griffiths, 2005). It is worth noticing that such a modeling

would be very complex in the context of dRSM. It would be probably easier to do it in the case of the SBM model first.

Extension of STBM to other types of edges Since the STBM model is the first one able to model a network with textual edges, this makes it a very useful tool in the analysis of social networks. Noticing that topic models have also been used for other types of data than texts, it may be possible to extend STBM to photos for instance. Indeed, Russell et al. (2006) have applied with success a topic model on images. Of course, the possibility to model interaction through images would have a direct application for studying social networks where photo sharing is possible (Flickr, Facebook, Tweeter). The extension of STBM to handle images in place of texts should direct. More interestingly, allowing to have both images and texts would require to find a way to model the correlation between the types of edges. Up to our knowledge, this is currently an open question.

Extension of STBM to dynamic networks Naturally, another perspective is to extend the STBM model to the dynamic framework. This would be possible by adding a state space model on the variables α and ρ, in order to model both the evolution of node connections and the evolution of topics. Such an approach has already been used by (Blei and Lafferty, 2006) for the Dynamic Topic Model, devoted to the text part. One expected difficulty is however the adaptation of our classification VEM algorithm in a such a context.

6.2.2 Application perspectives

Regarding the applications, we would like to go further in the analysis of the maritime flow network on the collapse of the Union of Soviet Socialist Republics. In order to extract more hidden information about this crisis we plan first to take only the direct ties in five different subgraphs instead of three. We will add two subgraphs to split more ports. For example "EASTERN BLOCK" that contains the ports strongly linked to the USSR and "OTHER SOCIALIST" which contains the ports from countries seen as socialist countries at some points in their history, but not anymore. Secondly, we plan to look at specific trades (e.g. bulks) and specific fleets (e.g. Soviet vs. capitalist), while considering categorical edges depending on ship's weight. Considering the maritime flow network in this way should allow us to discover finer groups with more specific functions.

Appendices

APPENDIX A

THE FORWARD-BACKWARD ALGORITHM

The forward-backward algorithm Let us begin by some conditions of independence using in this algorithm, such as:

$$
\begin{aligned}
p(X|Z_n) &= p(x_1,\ldots,x_n|Z_n)p(x_n+1,\ldots,x_N|Z_n) && \text{(A.1)}\\
p(x_1,\ldots,x_{n-1}|x_n,Z_n) &= \frac{p(Z_n,X_n|x_1,\ldots,x_{n-1})p(x_1,\ldots,x_{n-1})}{p(Z_n,x_n)} && \text{(A.2)}\\
&= \frac{p(X_n|Z_n,x_1,\ldots,x_{n-1})p(Z_n|x_1,\ldots,x_{n-1})p(x_1,\ldots,X_{n-1})}{p(x_n|Z_n)p(Z_n)} && \text{(A.3)}\\
&= p(x_1,\ldots,x_{n-1}|Z_n) && \text{(A.4)}\\
p(x_1,\ldots,x_{n-1}|Z_{n-1},Z_n) &= p(x_1,\ldots,x_{n-1}|Z_{n-1}) && \text{(A.5)}\\
p(x_{n+1},\ldots,x_N|Z_n,Z_{n+1}) &= p(x_{n+1},\ldots,x_N|Z_{n+1}) && \text{(A.6)}\\
p(x_{n+2},\ldots,x_N|Z_{n+1},X_{n+1}) &= p(x_{n+2},\ldots,x_N|Z_{n+1}) && \text{(A.7)}\\
p(X|Z_{n-1},Z_n) &= p(x_1,\ldots,x_{n-1}|Z_{n-1})p(x_n|Z_n)p(x_n+1,\ldots,x_N|Z_n) && \text{(A.8)}\\
p(x_{N+1}|X,Z_{N+1}) &= p(x_{N+1}|Z_{N+1}). && \text{(A.9)}
\end{aligned}
$$

Now, in order to find the posterior distribution of Z_n, we used the conditional independence property (A.1) to obtain decomposition of $p(Z_n|X)$ as seen below

:

$$\begin{aligned}
\gamma(Z_n) &= p(Z_n|X) \\
&= \frac{p(Z_n, X)}{p(X)} \\
&= \frac{p(X|Z_n)p(Z_n)}{p(X)} \\
&= \frac{p(x_1, \ldots, x_n|Z_n)p(x_n+1, \ldots, x_N|Z_n)p(Z_n)}{p(X)} \\
&= \frac{p(x_1, \ldots, x_n, Z_n)p(x_n+1, \ldots, x_N|Z_n)}{p(X)} \\
&= \frac{\alpha(Z_n)\beta(Z_n)}{p(X)},
\end{aligned}$$

where

$$\alpha(Z_n) \equiv p(x_1, \ldots, x_n, Z_n)$$

and

$$\beta(Z_n) \equiv p(x_n+1, \ldots, x_N|Z_n).$$

First, $\alpha(Z_n)$ is the joint probability of all observations up to time n and the value of Z_n. Using (A.4) and (A.5) of conditional independence to reformulate $\alpha(Z_n)$ as follows:

$$\begin{aligned}
\alpha(Z_n) &= p(x_1, \ldots, x_n, Z_n) && \text{(A.10)} \\
&= p(x_1, \ldots, x_n|Z_n)p(Z_n) && \text{(A.11)} \\
&= p(x_1, \ldots, x_{n-1}|Z_n)p(x_n|Z_n)p(Z_n) && \text{(A.12)} \\
&= p(x_1, \ldots, x_{n-1}, Z_n)p(x_n|Z_n) && \text{(A.13)} \\
&= p(x_n|Z_n) \sum_{Z_{n-1}} p(x_1, \ldots, x_{n-1}, Z_{n-1}, Z_n) && \text{(A.14)} \\
&= p(x_n|Z_n) \sum_{Z_{n-1}} p(x_1, \ldots, x_{n-1}Z_n|Z_{n-1})p(Z_{n-1}) && \text{(A.15)} \\
&= p(x_n|Z_n) \sum_{Z_{n-1}} p(x_1, \ldots, x_{n-1}|Z_{n-1})p(Z_n|Z_{n-1})p(Z_{n-1}) && \text{(A.16)} \\
&= p(x_n|Z_n) \sum_{Z_{n-1}} p(x_1, \ldots, x_{n-1}, Z_{n-1})p(Z_n|Z_{n-1}) && \text{(A.17)} \\
&= p(x_n|Z_n) \sum_{Z_{n-1}} \alpha(Z_{n-1})p(Z_n|Z_{n-1}). && \text{(A.18)}
\end{aligned}$$

The formulate (A.18) presents the forward recursion equation for $\alpha(Z_n)$ which is a set of K numbers, allowing us to express the joint probability $\alpha(Z_n)$

in terms of $\alpha(Z_{n-1})$. In order to start this recursion, we initialize the first term which is given by:

$$\alpha(Z_1) = p(x_1, z_1) = p(Z_1)p(x_1|Z_1).$$

We note that the normalization of $\alpha(Z_n)$ is presented by $\hat{\alpha}(Z_n)$ and given by:

$$\begin{aligned}\hat{\alpha}(Z_n) &= p(Z_n|x_1, \ldots, x_n) \\ &= \frac{p(x_1, \ldots, x_n, Z_n)}{p(x_1, \ldots, x_n)} \\ &= \frac{\alpha(Z_n)}{p(x_1, \ldots, x_n)}.\end{aligned}$$

At this point, we introduce the scaling factor c_n defined by conditional distributions over the observed variables, such that:

$$c_n = p(x_n|x_1, \ldots, x_{n-1}),$$

therefore, $\hat{\alpha}(Z_n) = \dfrac{\alpha(Z_n)}{\prod_{i=1}^{n} c_i}$, this relation gives rise to another recursion equation form:

$$c_n\alpha(Z_n) = p(x_n|Z_n) \sum_{Z_{n-1}} \hat{\alpha}(Z_{n-1})p(Z_n|Z_{n-1}), \tag{A.19}$$

note that,

$$\hat{\alpha}(Z_{n-1}) = \frac{\alpha(Z_{n-1})}{p(x_1, \ldots, x_{n-1})}.$$

Second, $\beta(Z_n)$ is also a set of K numbers, represents the conditional probability of all future data from time $n+1$ up to N given the value of Z_n. Similarly, we can find a recursion relation for these quantities, using the conditions (1.9) and (1.10) to find the following forms:

$$\begin{aligned}\beta(Z_n) &= p(x_n + 1, \ldots, x_N|Z_n) && \text{(A.20)} \\ &= \sum_{Z_{n+1}} p(x_{n+1}, \ldots, x_N, Z_{n+1}|Z_n) && \text{(A.21)} \\ &= \sum_{Z_{n+1}} p(x_{n+1}, \ldots, x_N|Z_{n+1}, Z_n)p(Z_{n+1}|Z_n) && \text{(A.22)} \\ &= \sum_{Z_{n+1}} p(x_{n+1}, \ldots, x_N|Z_{n+1})p(Z_{n+1}|Z_n) && \text{(A.23)} \\ &= \sum_{Z_{n+1}} p(x_{n+2}, \ldots, x_N|Z_{n+1})p(X_{n+1}|Z_{n+1})p(Z_{n+1}|Z_n) && \text{(A.24)} \\ &= \sum_{Z_{n+1}} \beta(Z_{n+1})p(X_{n+1}|Z_{n+1})p(Z_{n+1}|Z_n). && \text{(A.25)}\end{aligned}$$

Conversely to $\alpha(Z_n)$, $\beta(Z_n)$ have a backward message that allows us to evaluate $\beta(Z_n)$ in terms of $\beta(Z_{n-1})$. Again, to start this recursion we can start with $n = N$ and initialize $\beta(Z_N) = 1$.

We define, the ratio of two conditional probabilities noted $\hat{\beta}(Z_n)$, such that:

$$\hat{\beta}(Z_n) = \frac{p(x_{n+1}, \ldots, x_N | Z_n)}{p(x_{n+1}, \ldots, x_N | x_1, \ldots, x_n)},$$

where, we can also redefine $\beta(Z_n)$ using the scale c_n and $\hat{\beta}(Z_n)$ as follows:

$$\beta(Z_n) = \hat{\beta}(Z_n) \prod_{i=n+1}^{N} c_i,$$

which leads to,

$$c_{n+1}\hat{\beta}(Z_n) = \sum_{Z_{n+1}} \hat{\beta}(Z_{n+1}) p(x_{n+1}|Z_{n+1}) p(Z_{n+1}|Z_n). \tag{A.26}$$

Ultimately, using $\hat{\alpha}(Z_n)$ and $\hat{\beta}(Z_n)$ to find the following form to $\gamma(Z_n)$,

$$\gamma(Z_n) = \frac{\hat{\alpha}(Z_n) \prod_{i=1}^{n} c_i \hat{\beta}(Z_n) \prod_{i=n+1}^{N} c_i}{\prod_{i=1}^{N} c_i} = \hat{\alpha}(Z_n)\hat{\beta}(Z_n).$$

APPENDIX B

DERIVATION OF THE LOWER BOUND

In the following, we denote $x^{(t)} = \frac{\sum_{s=1}^{S} \hat{\gamma}_s^{(t)}}{S}$ an observed variable. The lower bound is given by:

$$\begin{aligned}
\tilde{\mathcal{L}}(q,\theta,\xi) &= \sum_Z \int_\gamma \int_\nu q(Z,\gamma,\nu) \log \frac{p(X|Z,\Pi)h(Z,\gamma,\xi)p(\gamma|\nu,\Sigma)p(\nu|\mu_0,A,\Phi,V_0)}{q(Z,\gamma,\nu)} d\nu d\gamma \\
&= E_{Z,\gamma,\nu}[\log \frac{p(X|Z,\Pi)h(Z,\gamma,\xi)p(\gamma|\nu,\Sigma,B)p(\nu|\mu_0,A,\Phi,V_0)}{q(\gamma)q(\nu)\prod_{i=1}^{N} q(Z_i)}] \\
&= E_Z(\log p(X|Z,\Pi)) + E_{Z,\gamma}(\log h(Z,\gamma,\xi)) + E_{\gamma,\nu}(\log p(\gamma|\nu,\Sigma,B)) \\
&+ E_\nu(\log p(\nu|\mu_0,A,\Phi,V_0)) - E_\gamma(\log q(\gamma)) - E_\nu(\log q(\nu)) - E_Z(\log(\prod_{i=1}^{N} q(Z_i))).
\end{aligned}$$

Note that (see Proposition 3.3.3),

$$q(\nu) \propto p(\nu^{(1)}|\mu_0,V_0)\Big[\prod_{t=2}^{T} p(\nu^{(t)}|\nu^{(t-1)},A,\Phi)\Big]\Big[\prod_{t=1}^{T} \mathcal{N}(\frac{\sum_{s=1}^{S} \hat{\gamma}_s^{(t)}}{S}; B\nu^{(t)}, \frac{\Sigma}{S})\Big].$$

As pointed out in this proposition, this corresponds to the form of the posterior distribution associated with a state space model with parameter $\theta^{'}$ and with observed outputs $x = (x^{(t)})_t$. If we denote $p(x|\theta^{'})$ the likelihood associated with this model, and the joint likelihood $p(x,\nu|\theta^{'})$, we have

$$q(\nu) = \frac{p(x,\nu|\theta^{'})}{p(x|\theta^{'})}.$$

Therefore

$$E_\nu(\log q(\nu)) = E_\nu(\log p(\nu|\mu_0, A, \Phi, V_0)) + E_\nu(\log p(\frac{\sum_{s=1}^{S}\hat{\gamma}_s^{(t)}}{S}|\nu^{(t)}, \frac{\Sigma}{S}, B)) - \log p(x|\theta^{'}).$$

This leads to,

$$E_\nu(\log p(\nu|\mu_0, A, \Phi, V_0)) - E_\nu(\log q(\nu)) = -E_\nu(\log p(\frac{\sum_{s=1}^{S}\hat{\gamma}_s^{(t)}}{S}|\nu^{(t)}, \frac{\Sigma}{S}, B)) + \log p(x|\theta^{'}),$$

and $\tilde{\mathcal{L}}(q, \theta, \xi)$ can be written as follows:

$$\begin{aligned}
\tilde{\mathcal{L}}(q,\theta,\xi) &= \sum_Z \int_\gamma \int_\nu q(Z,\gamma,\nu) \log \frac{p(X|Z,\Pi)h(Z,\gamma,\xi)p(\gamma|\nu,\Sigma)p(\nu|\mu_0,A,\Phi,V_0)}{q(Z,\gamma,\nu)} \\
&= E_{Z,\gamma,\nu}[\log \frac{p(X|Z,\Pi)h(Z,\gamma,\xi)p(\gamma|\nu,\Sigma,B)p(\nu|\mu_0,A,\Phi,V_0)}{q(\gamma)q(\nu)\prod_{i=1}^{N} q(Z_i)}] \\
&= E_Z(\log p(X|Z,\Pi)) + E_{Z,\gamma}(\log h(Z,\gamma,\xi)) + E_{\gamma,\nu}(\log p(\gamma|\nu,\Sigma,B)) \\
&- E_\gamma(\log q(\gamma)) - E_\nu(\log p(\frac{\sum_{s=1}^{S}\hat{\gamma}_s^{(t)}}{S}|\nu^{(t)}, \frac{\Sigma}{S}, B)) - E_Z(\log(\prod_{i=1}^{N} q(Z_i))) \\
&+ \log p(x|\theta^{'}).
\end{aligned}$$

We explicit below each of the terms of the bound $\tilde{\mathcal{L}}(q, \theta)$.

1. $E_Z(\log p(X|Z,\Pi))$:

$$\begin{aligned}
E_Z(\log p(X|Z,\Pi)) &= \sum_{t=1}^{T}\sum_{q,l}^{Q}\sum_{c=0}^{C}\sum_{i\neq j}^{N} E_z(\delta(X_{ij}^{(t)} = c) Z_{iq}^{(t)} Z_{jl}^{(t)} \log(\Pi_{ql}^{c}) \\
&= \sum_{t=1}^{T}\sum_{q,l}^{Q}\sum_{c=0}^{C}\sum_{i\neq j}^{N} \delta(X_{ij}^{(t)} = c) \tau_{iq}^{(t)} \tau_{jl}^{(t)} \log(\Pi_{ql}^{c})
\end{aligned}$$

2. $E_{Z,\gamma}(\log h(Z,\gamma,\xi))$:

$$\begin{aligned}
E_{Z,\gamma}(\log h(Z,\gamma,\xi)) &= E_{Z,\gamma}\Big[\sum_{t=1}^{T}\sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{s=1}^{S} y_{is} Z_{iq}^{(t)} \Big(\gamma_{sq}^{(t)} - \big(\xi_s^{-1(t)} \sum_l \exp(\gamma_{sq}^{(t)}) \\
&- 1 + \log(\xi_s^{(t)})\big)\Big)\Big] \\
&= \sum_{t=1}^{T}\sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{s=1}^{S} y_{is}\Big(\tau_{iq}^{(t)}\hat{\gamma}_{sq}^{(t)} - \tau_{iq}^{(t)}\xi_s^{-1(t)} \sum_{l=1}^{Q} \exp(\hat{\gamma}_{sl}^{(t)} + \frac{\hat{\sigma}_{sl}^{2^{(t)}}}{2}) \\
&+ \tau_{iq}^{(t)} - \tau_{iq}^{(t)} \log(\xi_s^{(t)})\Big) \\
&= \sum_{t=1}^{T}\sum_{s=1}^{S}\Big(r_s^{(t)}\hat{\gamma}_{sq}^{(t)} - N_s\xi_s^{-1(t)} \sum_{l=1}^{Q} \exp(\hat{\gamma}_{sl}^{(t)} + \frac{\hat{\sigma}_{sl}^{2^{(t)}}}{2}) + N_s - N_s \log(\xi_s^{(t)})\Big)
\end{aligned}$$

where denote $r_s^{(t)}$ is a quantity $\sum_{i=1}^{N} \tau_{iq}^{(t)} y_{is}$.

3. $E_{\gamma,\nu}(\log p(\gamma|\nu, \Sigma, B))$:

$$\begin{aligned} E_{\gamma,\nu}(\log p(\gamma|\nu, \Sigma, B)) &= E_{\gamma,\nu}\Big(\log \prod_{t=1}^{T} \prod_{s=1}^{S} \mathcal{N}(\gamma_s^{(t)}; B\nu_s^{(t)}, \Sigma)\Big) \\ &= \sum_{t=1}^{T} \sum_{s=1}^{S} \Big(\log \mathcal{N}(\hat{\gamma}_s^{(t)}, B\hat{\nu}_s^{(t)}, \Sigma) - \frac{1}{2} tr(\Sigma^{-1} B^T \hat{V}^{(t)} B) - \frac{1}{2} tr(\Sigma^{-1} \hat{\sigma}_s^{(t)^2}) \Big) \end{aligned}$$

4. $E_{\gamma}(\log q(\gamma))$:

$$\begin{aligned} E_{\gamma}(\log q(\gamma)) &= E_{\gamma}\Big(\prod_{t=1}^{T} \prod_{s=1}^{S} \prod_{q=1}^{Q} \mathcal{N}(\gamma_{sq}^{(t)}; \hat{\gamma}_{sq}^{(t)}, \hat{\sigma}_{sq}^{2^{(t)}}) \Big) \\ &= \sum_{t=1}^{T} \sum_{s=1}^{S} \sum_{q=1}^{Q} -\log \Big((2\pi)^{\frac{1}{2}} \hat{\sigma}_{sq}^{(t)} \Big) - \frac{TQS}{2}. \end{aligned}$$

5. $E_{\nu}(\log p(\frac{\sum_{s=1}^{S} \hat{\gamma}_s^{(t)}}{S} | \nu^{(t)}, \frac{\Sigma}{S}, B))$:

$$E_{\nu}(\log p(\frac{\sum_{s=1}^{S} \hat{\gamma}_s^{(t)}}{S} | \nu^{(t)}, \frac{\Sigma}{S}, B)) = \sum_{t=1}^{T} \Big(\log \mathcal{N}(x^{(t)}; B\hat{\nu}^{(t)}, \Sigma/S) - \frac{1}{2} tr(\Sigma^{-1} S B^T \hat{V}^{(t)} B) \Big).$$

6. $E_Z(\log(\prod_{i=1}^{T} q(Z_i)))$:

$$\begin{aligned} E_Z(\log(\prod_{i=1}^{T} q(Z_i))) &= \sum_{i=1}^{N} E_Z(\log q(Z_i)) \\ &= \sum_{i=1}^{N} E_Z(\sum_{t=1}^{T} \sum_{q=1}^{Q} Z_{iq}^{(t)} \log(\tau_{iq})) \\ &= \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{q=1}^{Q} \tau_{iq}^{(t)} \log(\tau_{iq}^{(t)}). \end{aligned}$$

BIBLIOGRAPHY

A. Ahmed and E. P. Xing. On tight approximate inference of logistic-normal admixture model. *In Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 1–8, 2007.

E. M. Airoldi, D. M. Blei, S. E. Fienberg, E. P. Xing, and T. Jaakkola. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the international biometrics society annual meeting*, pages 1–34, 2006.

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014, 2008.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, pages 267–281, 1973.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

R. Albert and A. L. Barabási. Statistical mechanics of complex networks. *Modern Physics*, 74:47–97, 2002.

R. Albert, H. Jeong, and A. L. Barabasi. Diameter of the world-wide web. *Nature*, 401:130–131, 1999.

C. Ambroise and C. Matias. New consistent and asymptotically normal parameter estimates for random-graph mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):3–35, 2012.

C. Ambroise, G. Grasseau, M. Hoebeke, P. Latouche, V. Miele, and F. Picard. The mixer R package (version 1.8), 2010. *http://cran.r-project.org/web/packages/mixer/*.

A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

A. L. Barabási and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Rev. Genet*, 5:101–113, 2004.

P. J. Bickel and A. Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.

C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Machine Intel*, 7:719–725, 2000a.

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

C. Biernacki, G. Celeux, and G. Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics and Data Analysis*, 41(3-4):561–575, 2003.

N. Biggs, E. K. Lloyd, and R. J. Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1976.

J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4:126, 1998.

C. M. Bishop. *Pattern recognition and machine learning*. Springer-Verlag, 2006.

D. Blei and J. Lafferty. A correlated topic model of science. *Annals of Applied Statistics*, 1(1):17–35, 2007a.

D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

D. M. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007b.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

V. D. Blondel, J-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10:10008–10020, 2008.

C. Bouveyron, Y. Jernite, P. Latouche, and L. Nouedoui. The rambo R package (version 1.1), 2013. *http://cran.r-project.org/web/packages/Rambo/*.

C. Bouveyron, P. Latouche, and R. Zreik. The stochastic topic block model for the clustering of vertices in networks with textual edges. *Statistics and Computing*, page DOI, 2016.

A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33:309–320, 2000.

G. Celeux and G. Govaert. A classification em algorithm for clustering and two stochastic versions. *Computational Statistics Quaterly*, 2(1):73–82, 1991.

J. Chang and D. M. Blei. Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics*, pages 81–88, 2009.

E. Côme and P Latouche. Model selection and clustering in stochastic block models with the exact integrated complete data likelihood. *Statistical Modelling*, page doi: 10.1177/1471082X15577017, 2015.

E. Côme, N. A. Randriamanamihaga, L. Oukhellou, and P. Aknin. Spatio-temporal analysis of dynamic origin-destination data using latent dirichlet allocation: Application to vélib'bike sharing system of paris. In *TRB 93rd Annual meeting*, page 19p. TRANSPORTATION RESEARCH BOARD, 2014.

A. Corduneanu and C. M. Bishop. Variational bayesian model selection for mixture distributions. In *Artificial intelligence and Statistics*, volume 2001, pages 27–34. Morgan Kaufmann Waltham, MA, 2001.

A. Dasgupta and A. E. Raftery. Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93(441):294–302, 1998.

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

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

S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letter*, 85:4633–4636, 2000.

C. Dubois, C. T. Butts, and P. Smyth. Stochastic blockmodelling of relational event dynamics. In *International Conference on Artificial Intelligence and Statistics*, volume 31 of the Journal of Machine Learning Research Proceedings, pages 238–246, 2013.

C. Ducruet. Network diversity and maritime flows. *Journal of Transport Geography*, 30:77–88, 2013.

P. Erdös and A. Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

S. E. Fienberg and S. Wasserman. Categorical data analysis of single sociometric relations. *Sociological methodology*, 12:156–192, 1981a.

S. E. Fienberg and S. S. Wasserman. Categorical data analysis of single sociometric relations. *Sociological Methodology*, 12:156–192, 1981b.

J. R. Foulds, C. DuBois, A. U. Asuncion, C. T. Butts, and P Smyth. A dynamic relational infinite feature model for longitudinal social networks. In *International Conference on Artificial Intelligence and Statistics*, pages 287–295, 2011.

L. C. Freeman. Some antecedents of social network analysis. *Connections*, 19(1): 39–42, 1996.

A. E. Gelfand and A. FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410): 398–409, 1990.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

Z. Ghahramani and T. L. Griffiths. Infinite latent feature models and the indian buffet process. In *Advances in neural information processing systems*, pages 475–482, 2005.

M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821, 2002.

D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *Advances in social networks analysis and mining (ASONAM), 2010 international conference on*, pages 176–183. IEEE, 2010.

B. Grun and K. Hornik. The mixer topicmodels package (version 0.2-3), 2013. *http://cran.r- project.org/web/packages/topicmodels/*.

M. S. Handcock, A. E. Raftery, and J. M. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007.

A. C. Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, Cambridge, UK, 1989.

R. J. Hathaway. Another interpretation of the EM algorithm for mixture distributions. *Statistics & Probability Letters*, 4(2):53–56, 1986.

C. Heaukulani and Z. Ghahramani. Dynamic probabilistic models for latent feature propagation in social networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 275–283, 2013.

Q. Ho, L. Song, and E. P. Xing. Evolving cluster mixed-membership blockmodel for time-evolving networks. In *International Conference on Artificial Intelligence and Statistics*, pages 342–350, 2011.

P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97 (460):1090–1098, 2002.

J. M. Hofman and C. H. Wiggins. Bayesian approach to network modularity. *Physical review letters*, 100(25):258701, 2008.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

E. E. Holmes, E. J. Ward, and K. Wills. Marss: Multivariate autoregressive state-space models for analyzing time-series data. *The R Journal*, 4(1):11–19, 2012.

T. S. Jaakkola. 10 tutorial on variational approximation methods. *Advanced mean field methods: theory and practice*, page 129, 2001.

Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

Y. Jernite, P. Latouche, C. Bouveyron, P. Rivera, L. Jegou, and S. Lamassé. The random subgraph model for the analysis of an acclesiastical network in merovingian gaul. *Annals of Applied Statistics*, 8(1):55–74, 2014.

M. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

Leonard Kaufman and Peter J Rousseeuw. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis*, pages 68–125, 1990.

C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, pages 381–391, 2006.

M. Kim and J. Leskovec. Nonparametric multi-group membership model for dynamic networks. In *Advances in Neural Information Processing Systems (25)*, pages 1385–1393, 2013a.

M. Kim and J. Leskovec. Nonparametric multi-group membership model for dynamic networks. In *Advances in neural information processing systems*, pages 1385–1393, 2013b.

T. Krishnan and G. J. McLachlan. *The EM algorithm and extensions*. John Wiley, New York, 1997.

P. N. Krivitsky and M. Handcock. Fitting position latent cluster models for social networks with latentnet. 2008.

L. Labiod and Y. Bennani. A spectral based clustering algorith for categorical data with maximum modularity. In *ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium)*, 2011.

V. Lacroix, C.G. Fernandes, and M.-F. Sagot. Motif search in graphs: pplication to metabolic networks. *Transactions in Computational Biology and Bioinformatics*, 3:360–368, 2006.

J. D. Lafferty and D. M. Blei. Correlated topic models. In Y. Weiss, B. Schölkopf, and J.C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 147–154. MIT Press, 2006.

P. Latouche. *Modèles de graphes aléatoires à structure cachée pour l'analyse des réseaux*. PhD thesis, Citeseer, 2011.

P. Latouche, E Birmelé, and C. Ambroise. Overlapping stochastic block models with application to the french political blogosphere. *Annals of Applied Statistics*, 5(1):309–336, 2011.

P. Latouche, E Birmelé, and C. Ambroise. Variational bayesian inference and complexity control for stochastic block models. *Statistical Modelling*, 12(1): 93–115, 2012.

P. Latouche, E Birmelé, and C. Ambroise. Model selection in overlapping stochastic block models. *Electronic Journal of Statistics*, 8(1):762–794, 2014.

P. Latouche, R. Zreik, and C. Bouveyron. Cluster identification in maritime flows with stochastic methods. *Maritime Networks: Spatial Structures and Time Dynamics, Routledge*, 2015.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

B. G. Leroux. Consistent estimation of amixing distribution. *Annals of Statistics*, 20:1350–1360, 1992.

Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: joint models of topic and author community. In *proceedings of the 26th annual international conference on machine learning*, pages 665–672. ACM, 2009.

J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

M. Mariadassou, S. Robin, and C. Vacher. Uncovering latent structure in valued graphs: a variational approach. *Annals of Applied Statistics*, 4(2):715–742, 2010.

C. Matias and V. Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B.*, 2016.

C. Matias and S. Robin. Modeling heterogeneity in random graphs through latent space models: a selective review. *Esaim Proc. and Surveys*, 47:55–74, 2014.

A. Mc Daid, T. B. Murphy, Frieln N., and N. J. Hurley. Improved bayesian inference for the stochastic block model with application to large networks. *Computational Statistics and Data Analysis*, 60:12–31, 2013.

A. McCallum, A. Corrada-Emmanuel, and X. Wang. The author-recipient-topic model for topic and role discovery in social networks, with application to enron and academic email. In *Workshop on Link Analysis, Counterterrorism and Security*, pages 33–44, 2005.

G. McLachlan and D. Peel. *Finite mixture models*. John Wiley & Sons, 2004.

G. J. McLachlan and K. E. Basford. Mixture models. inference and applications to clustering. *Statistics: Textbooks and Monographs, New York: Dekker, 1988*, 1, 1988.

M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

T. P. Minka. From hidden markov models to linear dynamical systems. Technical report, MIT, 1998.

Tom Minka. From hidden markov models to linear dynamical systems. Technical report, Citeseer, 1999.

J. L. Moreno. *Who shall survive?: A new approach to the problem of human interrelations.* Nervous and Mental Disease Publishing Co, 1934.

R. M Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

D. Newman, P. Smyth, M. Welling, and A. U. Asuncion. Distributed inference for latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1081–1088, 2007.

M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review Letter E*, 69:0066133, 2004.

M. E. J Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455): 1077–1087, 2001.

G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435: 814–818, 2005.

C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the tenth ACM PODS*, pages 159–168. ACM, 1998.

N. Pathak, C. DeLong, A. Banerjee, and K. Erickson. Social topic models for community extraction. In *The 2nd SNA-KDD workshop*, volume 8. Citeseer, 2008.

D. Peel and G. J. McLachlan. Robust mixture modelling using the t distribution. *Statistics and computing*, 10(4):339–348, 2000.

L. R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, pages 846–850, 1971.

H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965a.

H. E. Rauch, F. Tung, and T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIASS Journal*, 3(8):1445–1450, 1965b.

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

F. Rossi, N. Villa-Vialaneix, and F. Hautefeuille. Exploration of a large database of French notarial acts with social network methods. *Digital Medievalist*, 9: 1–20, 7 2014.

B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1605–1614. IEEE, 2006.

M. Sachan, D. Contractor, T. Faruquie, and L. Subramaniam. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 331–340. ACM, 2012.

M. Salter-Townshend and T. B. Murphy. Variational bayesian inference for the latent position cluster model. In *Analyzing Networks and Learning with Graphs Workshop at 23rd annual conference on Neural Information Processing Systems (NIPS 2009), Whister, December 11 2009*, 2009.

P. Sarkar and A. W. Moore. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations Newsletter*, 7(2):31–40, 2005.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6: 461–464, 1978.

G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315. ACM, 2004.

Y. Sun, J. Han, J. Gao, and Y. Yu. itopicmodel: Information network-integrated topic modeling. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 493–502. IEEE, 2009.

M. Svensén and C. M. Bishop. Robust bayesian mixture modelling. *Neurocomputing*, 64:235–252, 2004.

M. Svensén and C. M. Bishop. Robust bayesian mixture modelling. *Neurocomputing*, 64:235–252, 2005.

Y. W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent Dirichlet allocation. *Advances in neural information processing systems*, 18:1353–1360, 2006.

Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82:8–19, 1987.

H. C. White, S. A. Boorman, and R. L. Breiger. Social structure from multiple networks. i. blockmodels of roles and positions. *American Journal of Sociology*, pages 730–780, 1976.

E. P. Xing, W. Fu, and L. Song. A state-space mixed membership blockmodel for dynamic network tomography. *The Annals of Applied Statistics*, 4(2):535–566, 2010.

K. S. Xu. Stochastic block transition models for dynamic networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1079–1087, 2015.

K. S. Xu and A. O. Hero III. Dynamic stochastic blockmodels: Statistical models for time-evolving networks. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 201–210. Springer, 2013.

T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin. Detecting communities and their evolutions in dynamic social networks—a bayesian approach. *Machine learning*, 82(2):157–189, 2011.

H. Zanghi, C. Ambroise, and V. Miele. Fast online graph clustering via erdos-renyi mixture. *Pattern recognition*, 41:3592–3599, 2008.

H. Zanghi, S. Volant, and C. Ambroise. Clustering based on random graph model embedding vertex features. *Pattern Recognition Letters*, 31(9):830–836, 2010.

D. Zhou, E. Manavoglu, J. Li, C. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *Proceedings of the 15th international conference on World Wide Web*, pages 173–182. ACM, 2006.

R. Zreik, P. Latouche, and C. Bouveyron. Classification automatique de réseaux dynamiques avec sous-graphes: étude du scandale enron. *Journal de la Société Française de Statistique*, 156(3):166–191, 2015.

R. Zreik, P. Latouche, and C. Bouveyron. The dynamic random subgraph model for the clustering of evolving networks. *Computational Statistics*, in press, 2016.